

FIG. 1

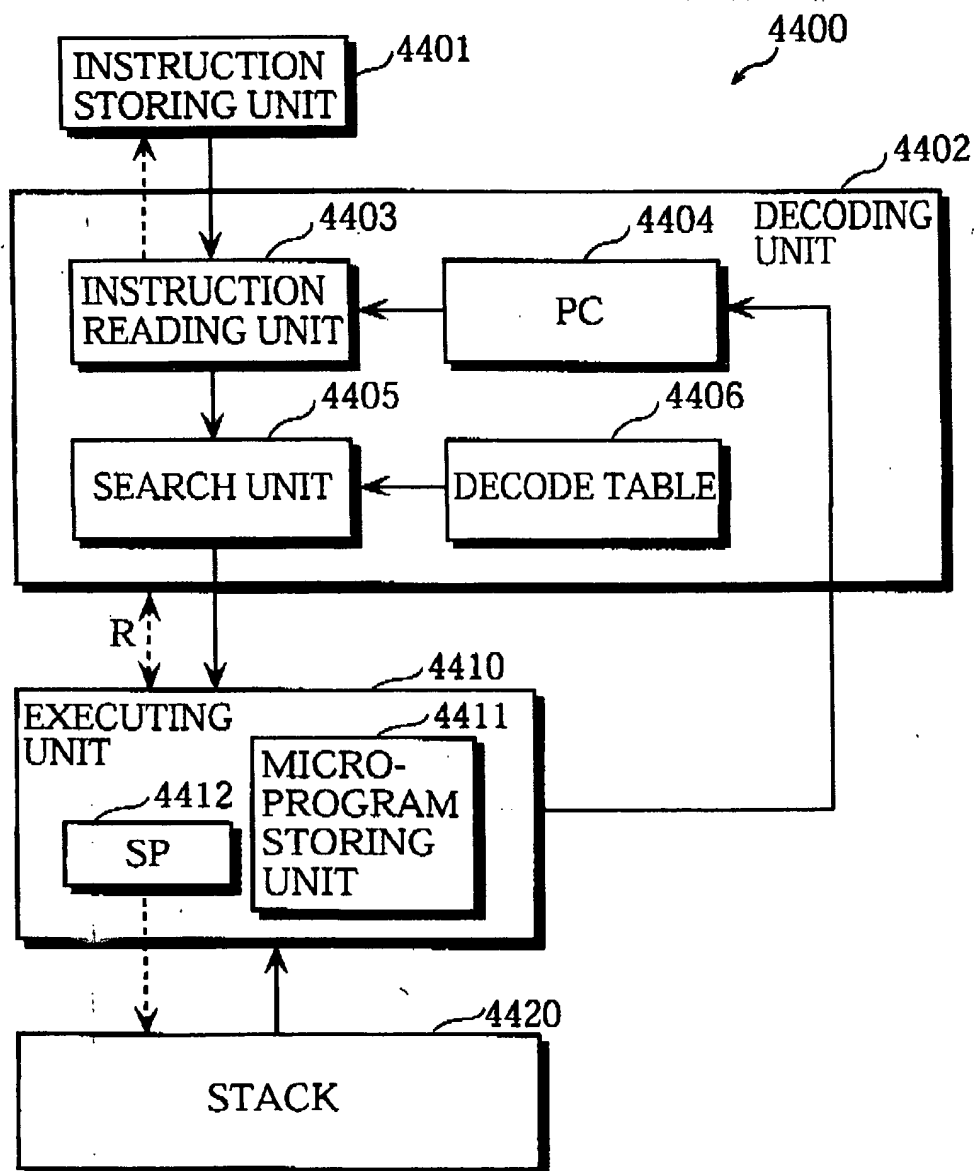






FIG. 4A

MICROPROGRAM FOR VIRTUAL MACHINE INSTRUCTION "Push"		
1:Inc	r3	; INCREMENT SP VALUE BY ONE
2:Load	r0,[r2]	; EXTRACT OPERAND AND ; PLACE IT ONTO REGISTER #0
3:Inc	r2	; INCREMENT VIRTUAL MACHINE PC BY ONE AND ; PREPARE FOR READING NEXT INSTRUCTION
4:Store	[r3],r0	; PUSH VALUE OF REGISTER #0 ONTO STACK
<MICROPROGRAM FOR JUMPING TO NEXT VIRTUAL MACHINE INSTRUCTION>		

FIG. 4B

MICROPROGRAM FOR VIRTUAL MACHINE INSTRUCTION "Add"		
1:Load	r0,[r3]	; EXTRACT VALUE FROM STACK ; PLACE IT ONTO REGISTER #0
2:Dec	r3	; DECREMENT VALUE OF VIRTUAL MACHINE SP BY ONE
3:Load	r1,[r3]	; EXTRACT VALUE FROM STACK ; PLACE IT ONTO REGISTER #1
4:Add	r0,r0,r1	; ADD VALUES OF REGISTER #0 AND #1 AND ; PLACE RESULT ONTO REGISTER #1
5:Store	[r3],r0	; PLACE VALUE OF REGISTER #0 ONTO STACK
<MICROPROGRAM FOR JUMPING TO NEXT VIRTUAL MACHINE INSTRUCTION>		

FIG. 4C

MICROPROGRAM FOR VIRTUAL MACHINE INSTRUCTION "Mult"		
1:Load	r0,[r3]	; EXTRACT VALUE FROM STACK AND ; PLACE IT ONTO REGISTER #0
2:Dec	r3	; DECREMENT VALUE OF VIRTUAL MACHINE SP BY ONE
3:Load	r1,[r3]	; EXTRACT VALUE FROM STACK AND ; PLACE IT ONTO REGISTER #1
4:Mult	r0,r0,r1	; MULTIPLY VALUES OF REGISTERS #0 AND #1 AND ; PLACE RESULT ONTO REGISTER #1
5:Store	[r3],r0	; PLACE VALUE OF REGISTER #0 ONTO STACK
<MICROPROGRAM FOR JUMPING TO NEXT VIRTUAL MACHINE INSTRUCTION>		

FIG. 4D

<MICROPROGRAM FOR JUMPING TO NEXT VIRTUAL MACHINE INSTRUCTION>		
1:Load	r0,[r2]	; READ VIRTUAL MACHINE INSTRUCTION (JUMP ADDRESS) INDICATED BY PC INTO REGISTER #0
2:Inc	r2	; INCREMENT VIRTUAL MACHINE PC VALUE BY ONE
3:Jmp	r0	; JUMP UNCONDITIONALLY TO LOCATION INDICATED BY REGISTER #0



FIG. 6

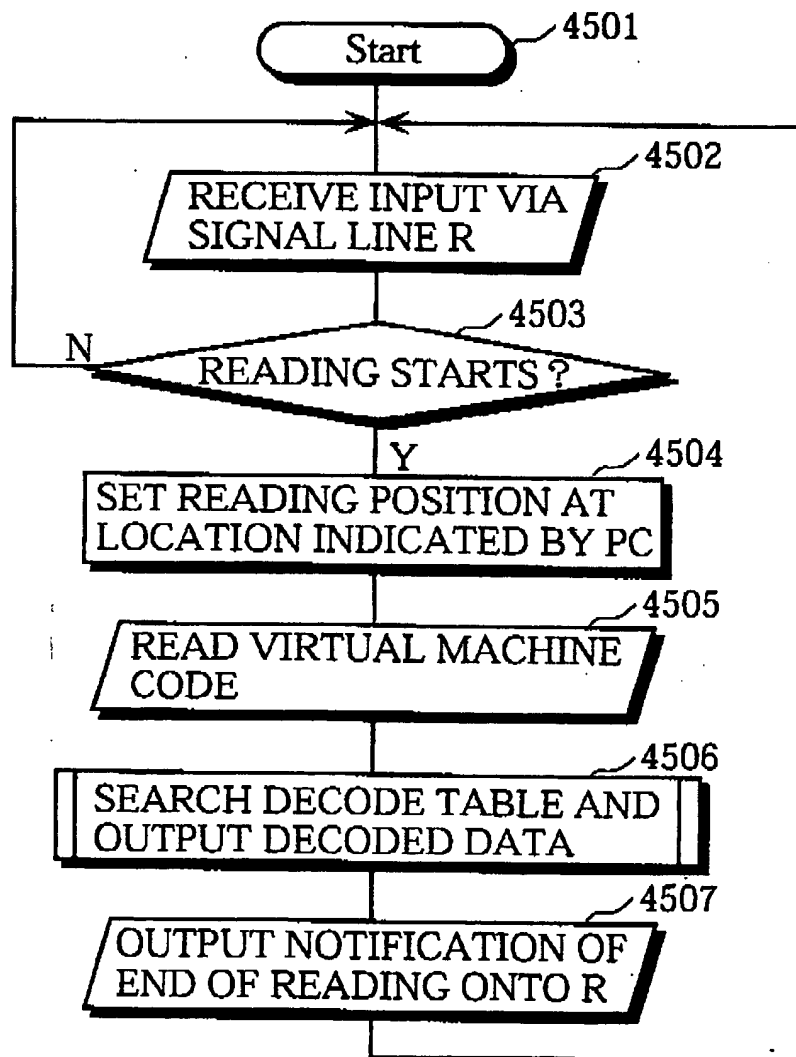


FIG. 7

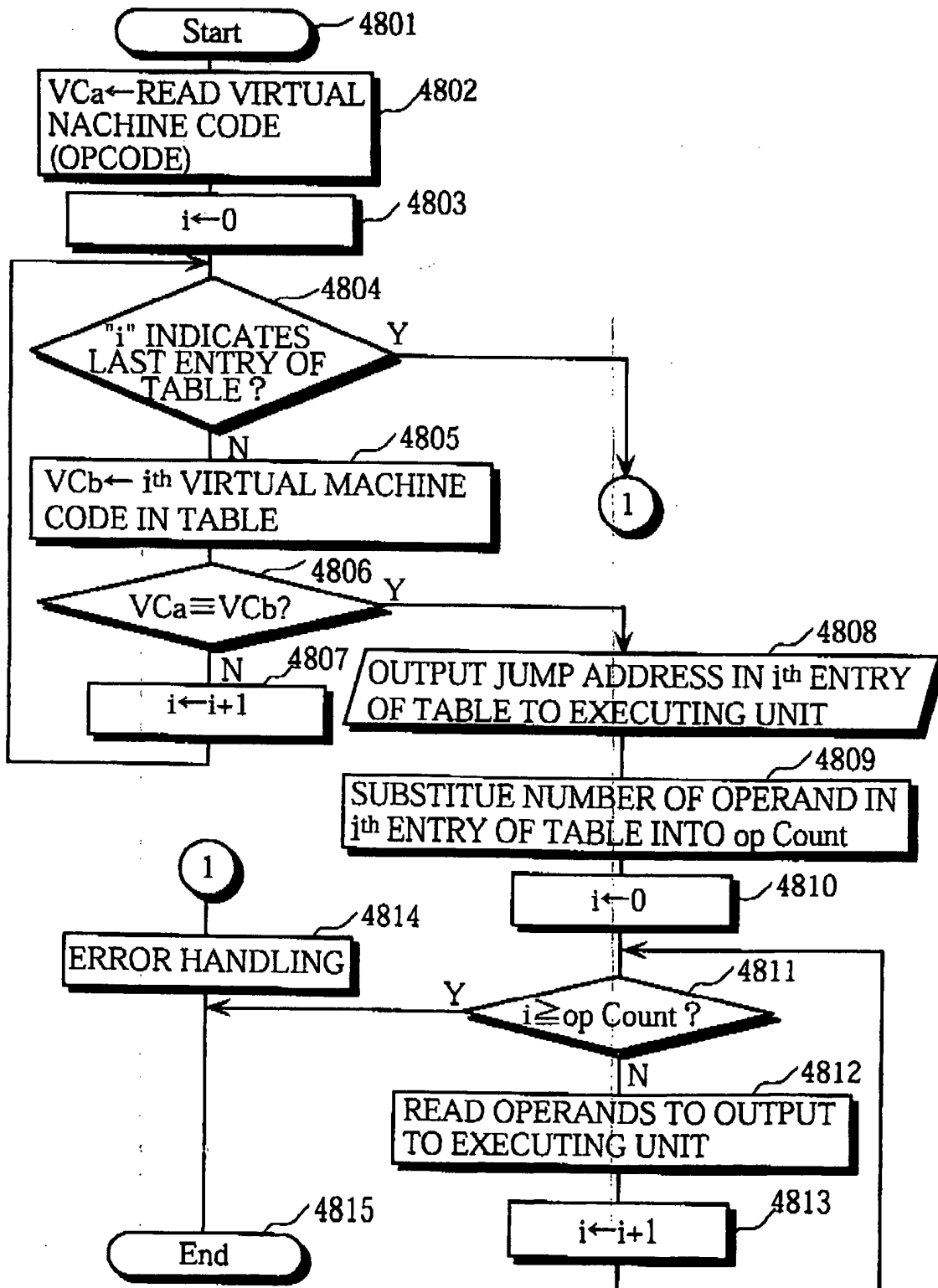


FIG. 8

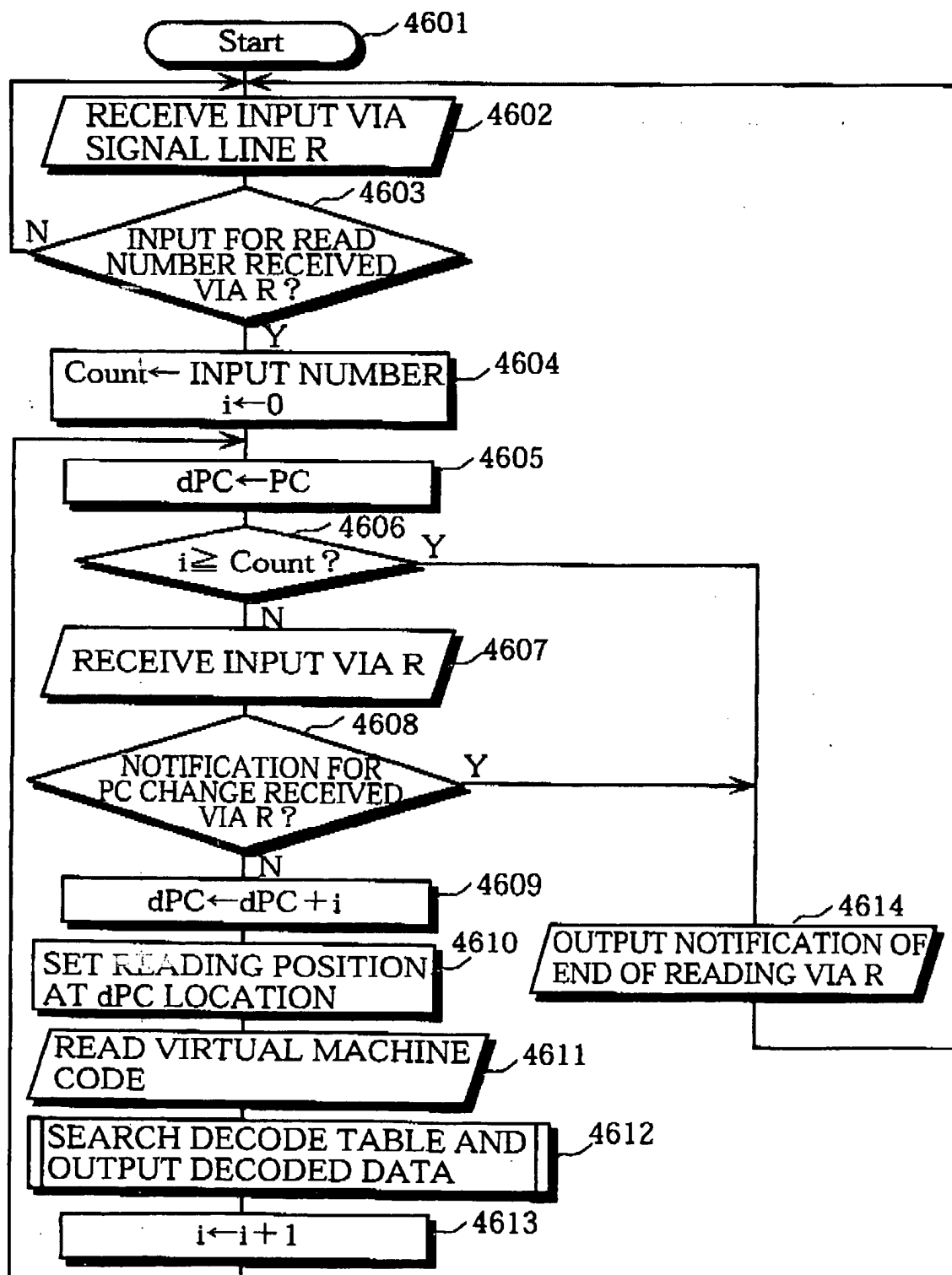




FIG. 9

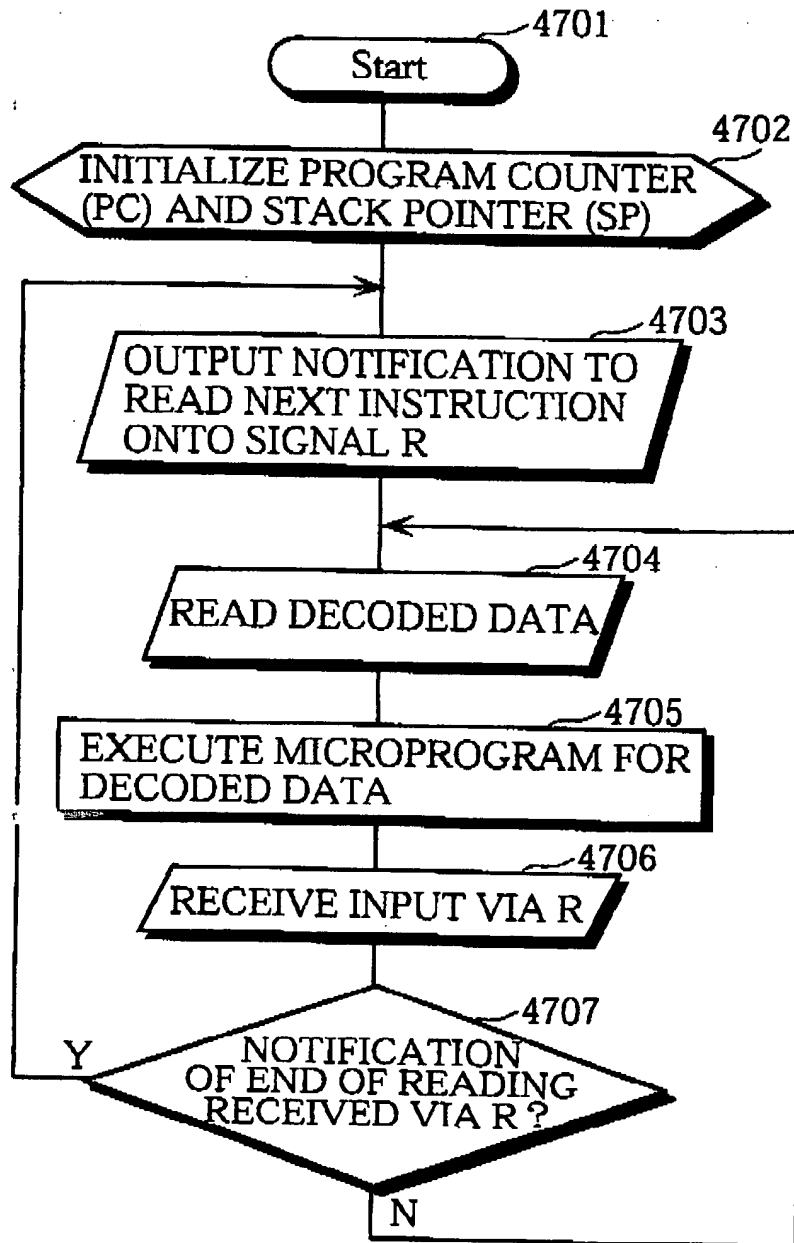


FIG. 10A

1:	Push
2:	2
3:	Push
4:	3
5:	Push
6:	4
7:	Add
8:	Mult
9:	Pop
10:	0

FIG. 10B

ARITHMETIC EXPRESSION :  $\langle \text{DATA AREA \#0} \rangle = 2 * (3 + 4)$

FIG. 10C

1:	$\langle \text{JUMP ADDRESS OF CODE TO PERFORM Push} \rangle$
2:	OPERAND "2"
3:	$\langle \text{JUMP ADDRESS OF CODE TO PERFORM Push} \rangle$
4:	OPERAND "3"
5:	$\langle \text{JUMP ADDRESS OF CODE TO PERFORM Push} \rangle$
6:	OPERAND "4"
7:	$\langle \text{JUMP ADDRESS OF CODE TO PERFORM Add} \rangle$
8:	$\langle \text{JUMP ADDRESS OF CODE TO PERFORM Mult} \rangle$
9:	$\langle \text{JUMP ADDRESS OF CODE TO PERFORM Pop} \rangle$
10:	OPERAND "0"

FIG. 11A

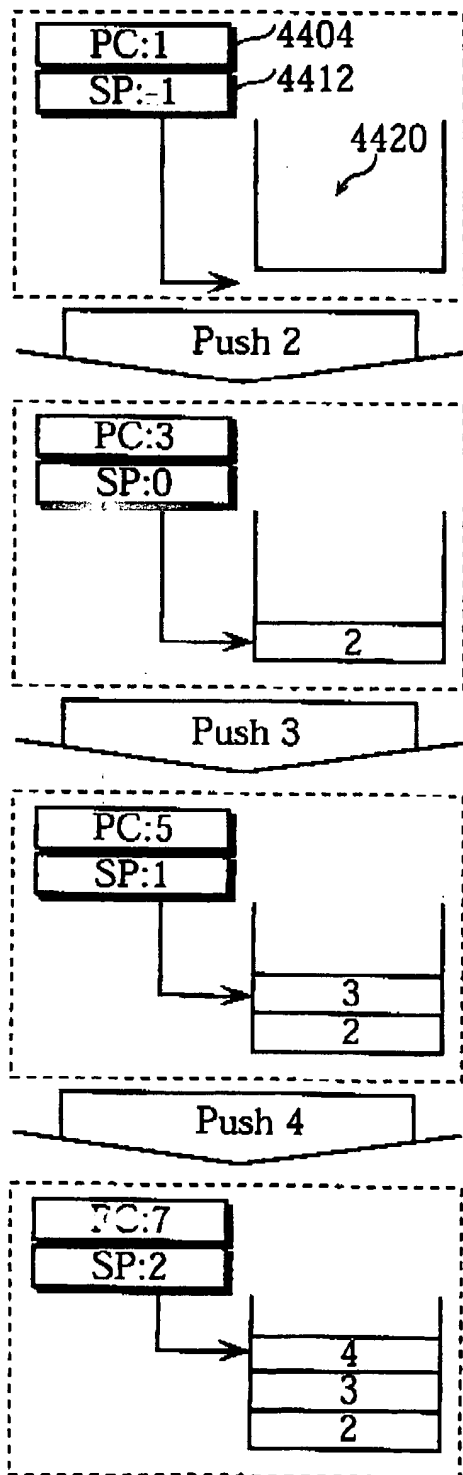


FIG. 11B

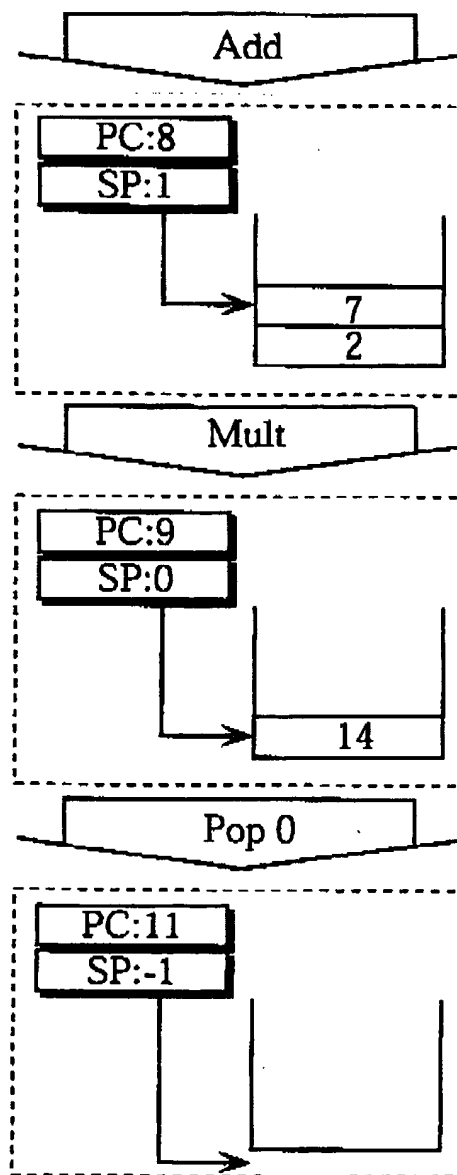


FIG. 12A

MICROPROGRAM FOR VIRTUAL MACHINE INSTRUCTION "Push"		
1:Inc	r3	: INCREMENT SP VALUE BY ONE
2:Store	[r3],r0	: PLACE VALUE OF TOS REGISTER (#0) : INTO STACK
3:Load	r0,[r2]	: EXTRACT OPERAND AND : PLACE IT ONTO TOS REGISTER
4:Inc	r2	: INCREMENT PC OF VIRTUAL MACHINE BY ONE TO PREPARE : FOR READING NEXT INSTRUCTION
<MICROPROGRAM FOR JUMPING TO NEXT VIRTUAL MACHINE INSTRUCTION>		

FIG. 12B

MICROPROGRAM FOR VIRTUAL MACHINE INSTRUCTION "Add"		
1:Load	r1,[r3]	: EXTRACT VALUE FROM STACK : PLACE IT ONTO REGISTER #1
2:Dec	r3	: DECREMENT VALUE OF VIRTUAL MACHINE PC BY ONE
3:Add	r0,r0,r1	: ADD VALUES OF REGISTERS #0 AND #1 AND : PLACE RESULT ONTO TOS REGISTER
<MICROPROGRAM FOR JUMPING TO NEXT VIRTUAL MACHINE INSTRUCTION>		

FIG. 12C

MICROPROGRAM FOR VIRTUAL MACHINE INSTRUCTION "Mult"		
1:Load	r1,[r3]	: EXTRACT VALUE FROM STACK AND : PLACE IT ONTO REGISTER #1
2:Dec	r3	: DECREMENT VALUE OF VIRTUAL MACHINE SP BY ONE
3:Mult	r0,r0,r1	: MULTIPLY VALUES OF REGISTERS #0 AND #1 AND : PLACE RESULT ONTO TOS REGISTER
<MICROPROGRAM FOR JUMPING TO NEXT VIRTUAL MACHINE INSTRUCTION>		

FIG. 12D

<MICROPROGRAM FOR JUMPING TO NEXT VIRTUAL MACHINE INSTRUCTION>		
1:Load	r1,[r2]	: READ VIRTUAL MACHINE INSTRUCTION (JUMP ADDRESS) : INDICATED BY PC INTO REGISTER #1
2:Inc	r2	: INCREMENT VIRTUAL MACHINE PC BY ONE
3:Jmp	r1	: JUMP UNCONDITIONALLY TO LOCATION INDICATED BY REGISTER #1

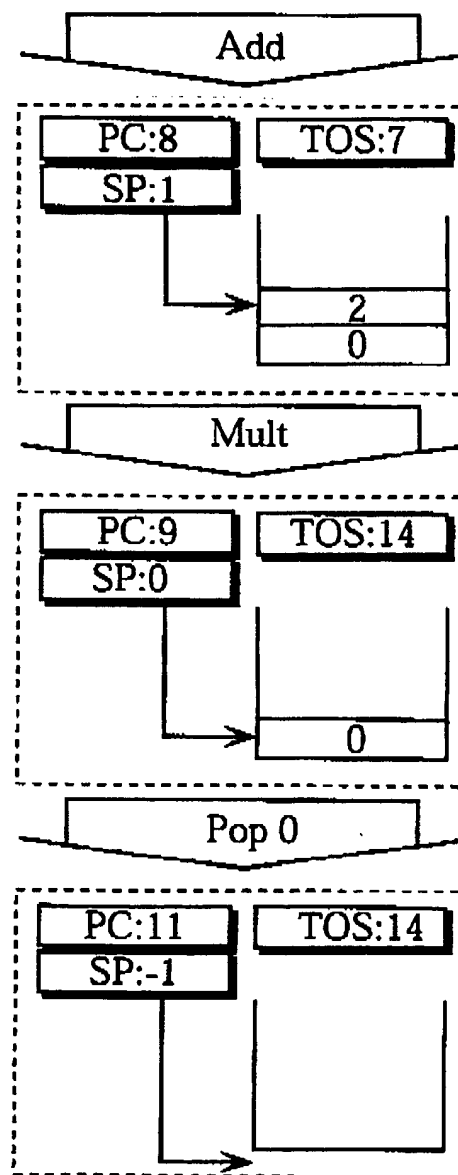
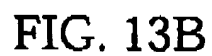
[illegible]

FIG. 14

STAGE NAME	NOTATION
INSTRUCTION FETCH	IF
INSTRUCTION DECODE AND REGISTER REFERENCE	RF
EXECUTION	ALU
MEMORY ACCESS	MEM
RESULT WRITING TO REGISTER	WB

FIG. 15

CLOCK	1	2	3	4	5	6	7	8
INSTRUCTION A	IF	RF	ALU	MEM	WB			
INSTRUCTION B		IF	RF	ALU	MEM	WB		
INSTRUCTION C			IF	RF	ALU	MEM	WB	
INSTRUCTION D				IF	RF	ALU	MEM	WB

FIG. 16

CLOCK	1	2	3	4	5	6	7	8
INSTRUCTION A1	IF	RF	ALU	MEM	WB			
INSTRUCTION A2	IF	RF	ALU	MEM	WB			
INSTRUCTION B1		IF	RF	ALU	MEM	WB		
INSTRUCTION B2		IF	RF	ALU	MEM	WB		
INSTRUCTION C1			IF	RF	ALU	MEM	WB	
INSTRUCTION C2			IF	RF	ALU	MEM	WB	
INSTRUCTION D1				IF	RF	ALU	MEM	WB
INSTRUCTION D2				IF	RF	ALU	MEM	WB

FIG. 17

CLOCK	1	2	3	4	5	6	7	8	9
INSTRUCTION A	IF	RF	ALU	MEM	WB				
INSTRUCTION B		IF	RF	•	ALU	MEM	WB		
INSTRUCTION C			IF	•	RF	ALU	MEM	WB	
INSTRUCTION D				•	IF	RF	ALU	MEM	WB

FIG. 18

CLOCK	1	2	3	4	5	6	7	8	9
INSTRUCTION A1	IF	RF	ALU	MEM	WB				
INSTRUCTION A2	IF	RF	ALU	MEM	WB				
INSTRUCTION B1		IF	RF	•	ALU	MEM	WB		
INSTRUCTION B2		IF	RF	ALU	MEM	WB			
INSTRUCTION C1			IF	RF	ALU	MEM	WB		
INSTRUCTION C2			IF	RF	•	ALU	MEM	WB	
INSTRUCTION D1				IF	RF	ALU	MEM	WB	
INSTRUCTION D2				IF	RF	•	ALU	MEM	WB

FIG. 19

CLOCK	1	2	3	4	5	6	7	8	9	10
INSTRUCTION A	IF	RF	ALU	MEM	WB					
INSTRUCTION B		IF	RF	•	•	ALU	MEM	WB		
INSTRUCTION C			IF	•	•	RF	ALU	MEM	WB	
INSTRUCTION D				•	•	IF	RF	ALU	MEM	WB

FIG. 20

CLOCK	1	2	3	4	5	6	7	8	9
INSTRUCTION A1	IF	RF	ALU	MEM	WB				
INSTRUCTION A2	IF	RF	ALU	MEM	WB				
INSTRUCTION B1		IF	RF	.	.	ALU	MEM	WB	
INSTRUCTION B2		IF	RF	ALU	MEM	WB			
INSTRUCTION C1			IF	RF	ALU	MEM	WB		
INSTRUCTION C2			IF	RF	ALU	MEM	WB		
INSTRUCTION D1				IF	RF	ALU	MEM	WB	
INSTRUCTION D2				IF	RF	.	ALU	MEM	WB

FIG. 21

CLOCK	1	2	3	4	5	6	7
INSTRUCTION A	IF	RF	ALU	MEM	WB		
INSTRUCTION B		IF	x				
INSTRUCTION C			IF	RF	ALU	MEM	WB

FIG. 22

CLOCK	1	2	3	4	5	6	7
INSTRUCTION A1	IF	RF	ALU	MEM	WB		
INSTRUCTION A2	IF	RF	ALU	MEM	WB		
INSTRUCTION B1		IF	x				
INSTRUCTION B2		IF	x				
INSTRUCTION C1			IF	RF	ALU	MEM	WB
INSTRUCTION C2			IF	RF	ALU	MEM	WB



FIG. 23

CLOCK	1	2	3	4	5	6	7	8	9	10	11
Load r1,[r2]	IF	RF	ALU	MEM	WB						
Inc r2		IF	RF	ALU	MEM	WB					
Jmp r1			IF	RF	ALU	MEM	WB				
				IF	x						
Load r1,[r3]					IF	RF	ALU	MEM	WB		
Dec r3						IF	RF	ALU	MEM	WB	
Mult r0,r0,r1							IF	RF	ALU	MEM	WB

FIG. 24

CLOCK	1	2	3	4	5	6	7	8	9	10	11
Load r1,[r2]	IF	RF	ALU	MEM	WB						
Inc r2	IF	RF	ALU	MEM	WB						
Jmp r1		IF	RF	•	ALU	MEM	WB				
		IF	RF	ALU	x						
			IF	RF	x						
			IF	RF	x						
				IF	x						
				IF	x						
Load r1,[r3]					IF	RF	ALU	MEM	WB		
Dec r3					IF	RF	ALU	MEM	WB		
Mult r0,r0,r1						IF	RF	•	ALU	MEM	WB





FIG. 27

0:Push	0	;i←0
2:Pop	[0]	
4:Push	0	;sum←0
6:Pop	[1]	
8:Push	[0]	;i<10?
10:Push	10	
12:Sub		
13:Brz	31	
15:Push	[1]	;sum←sum+i
17:Push	[0]	
19:Add		
20:Pop	[1]	
22:Push	[0]	;i←i+1
24:Push	1	
26:Add		
27:Pop	[0]	
29:Br	8	
31:Stop		

FIG. 28

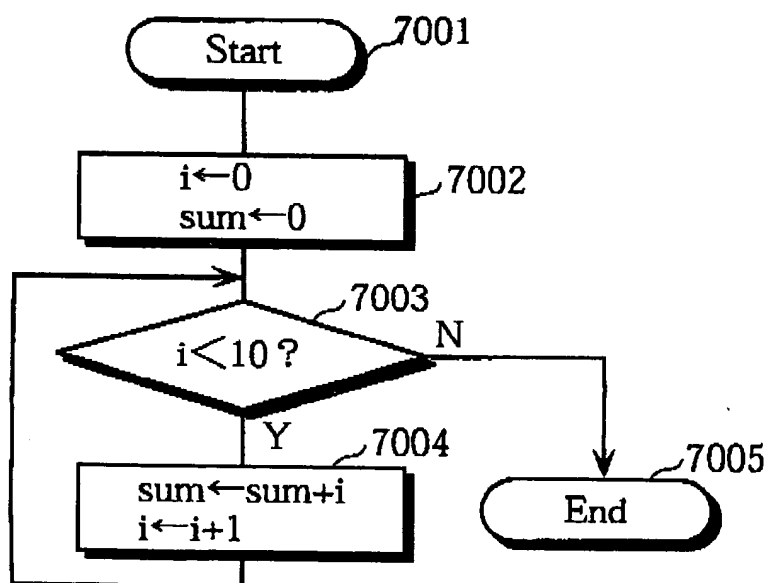




FIG. 30

VIRTUAL MACHINE CODE ADDRESS	VIRTUAL MACHINE CODE	REAL MACHINE CODE SIZE	CORRESPONDING REAL MACHINE CODE ADDRESS
0	Push 0	4	0-3
2	Pop [0]	5	4-8
4	Push 0	4	9-12
6	Pop [1]	5	13-17
8	Push [0]	5	18-22
10	Push 10	4	23-26
12	Sub	3	27-29
13	Brz 31	5	30-34
15	Push [1]	5	35-39
17	Push [0]	5	40-44
19	Add	3	45-47
20	Pop [1]	5	48-52
22	Push [0]	5	53-57
24	Push 1	4	58-61
26	Add	3	62-64
27	Pop [0]	5	65-69
29	Br 8	3	70-72
31	Stop	2	73-75

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

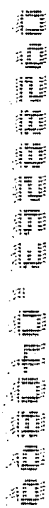
[illegible][illegible][illegible][illegible]



FIG. 32

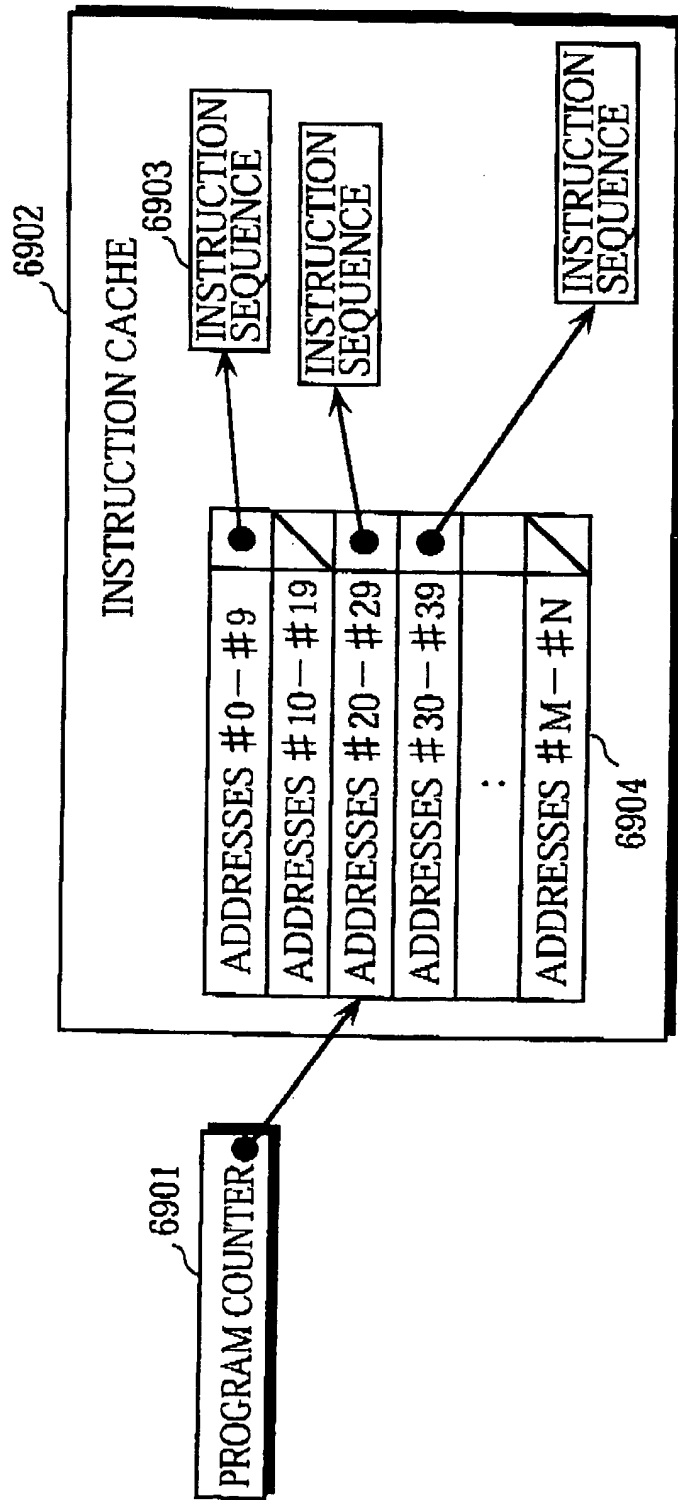


FIG. 33

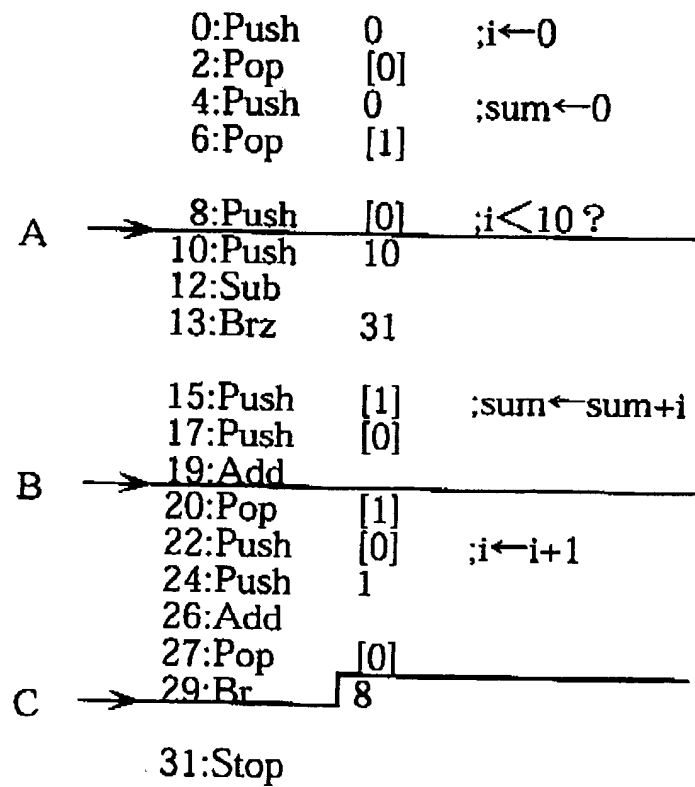




FIG. 35

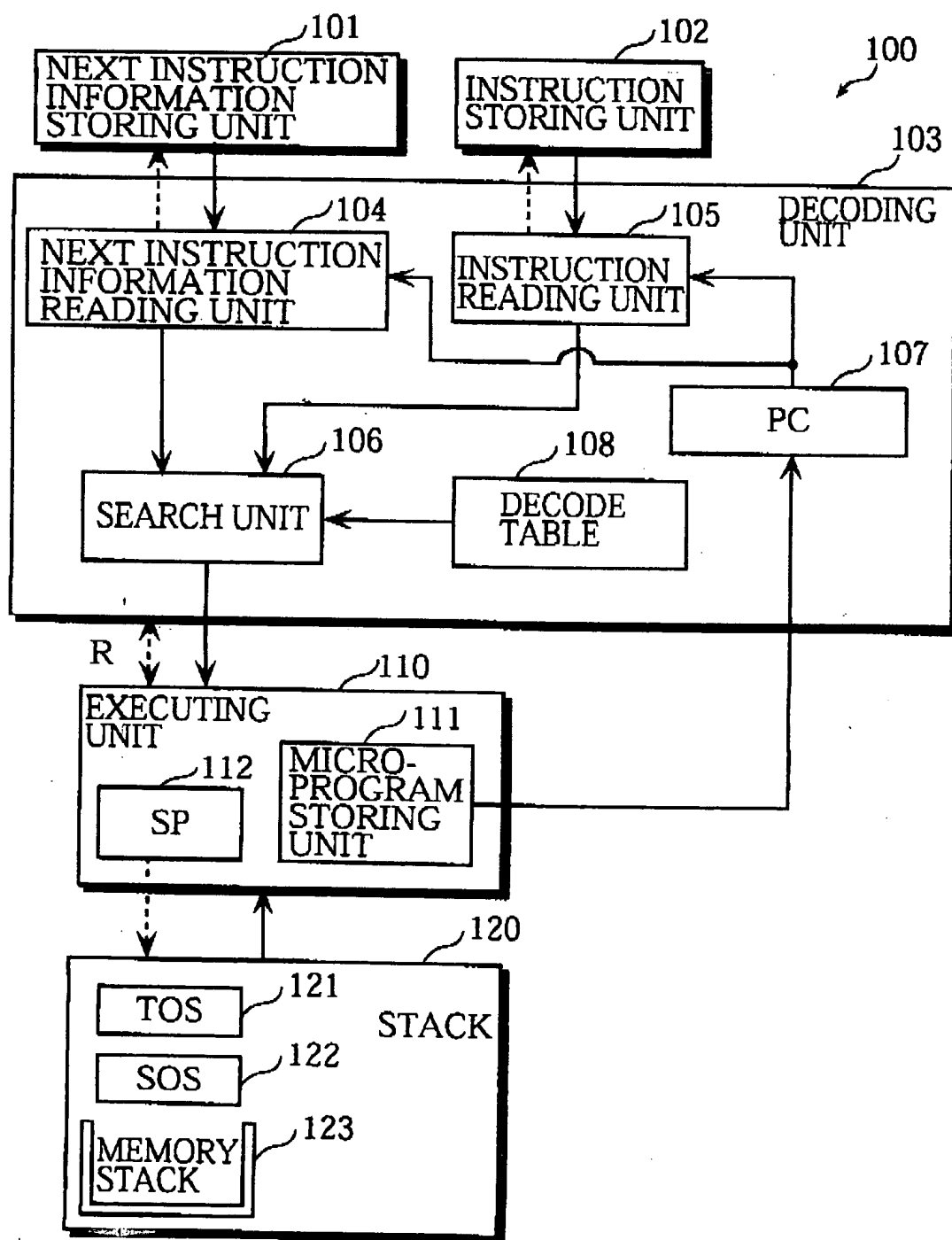


FIG. 36A

101

1:	U
2:	U
3:	U
4:	U
5:	D
6:	D
7:	D
8:	D
9:	U
10:	U

FIG. 36B

102

1:	Push
2:	2
3:	Push
4:	3
5:	Push
6:	4
7:	Add
8:	Mult
9:	Pop
10:	0



FIG. 38A

MICROPROGRAM FOR VIRTUAL MACHINE INSTRUCTION "Push" WITH "U"		
1:Load	r4,r0	;COPY VALUE OF TOS REGISTER (#0) INTO SOS REGISTER (#4)
2:Load	r0,[r2]	;READ OPERAND INTO TOS REGISTER
3:Inc	r2	;INCREMENT VIRTUAL MACHINE PC BY ONE TO PREPARE FOR READING NEXT INSTRUCTION
4:Inc	r3	;INCREMENT VIRTUAL MACHINE SP BY ONE
5:Store	[r3],r4	;PLACE SOS REGISTER VALUE INTO STACK
6:Load	r1,[r2]	;READ VIRTUAL MACHINE INSTRUCTION (JUMP ADDRESS) INDICATED BY PC INTO REGISTER #1
7:Inc	r2	;INCREMENT VIRTUAL MACHINE PC BY ONE
8:Jmp	r1	;JUMP UNCONDITIONALLY TO LOCATION INDICATED BY REGISTER #1

FIG. 38B

MICROPROGRAM FOR VIRTUAL MACHINE INSTRUCTION "Push" WITH "D"		
1:Load	r4,r0	;COPY VALUE OF TOS REGISTER (#0) INTO SOS REGISTER (#4)
2:Load	r0,[r2]	;READ OPERAND INTO TOS REGISTER
3:Inc	r2	;INCREMENT VIRTUAL MACHINE PC BY ONE TO PREPARE FOR READING NEXT INSTRUCTION
6:Load	r1,[r2]	;READ VIRTUAL MACHINE INSTRUCTION (JUMP ADDRESS) INDICATED BY PC INTO REGISTER #1
7:Inc	r2	;INCREMENT VIRTUAL MACHINE PC BY ONE
8:Jmp	r1	;JUMP UNCONDITIONALLY TO LOCATION INDICATED BY REGISTER #1











FIG. 43

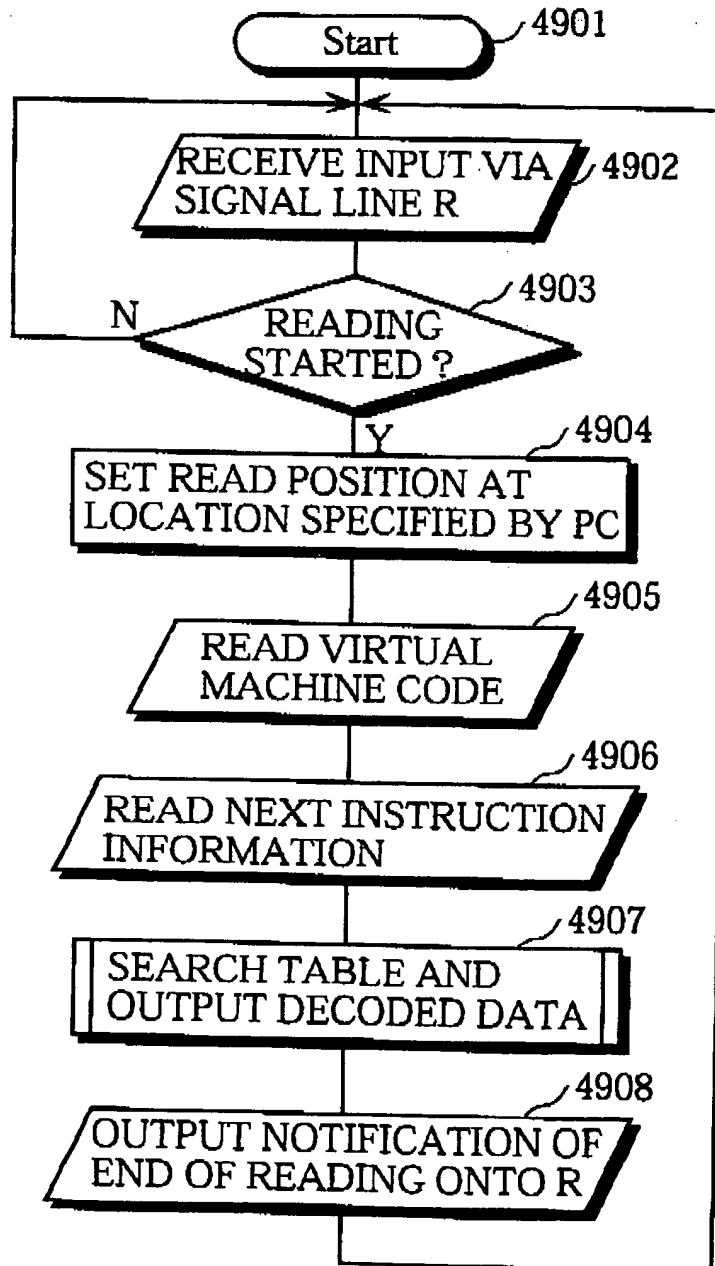


FIG. 44

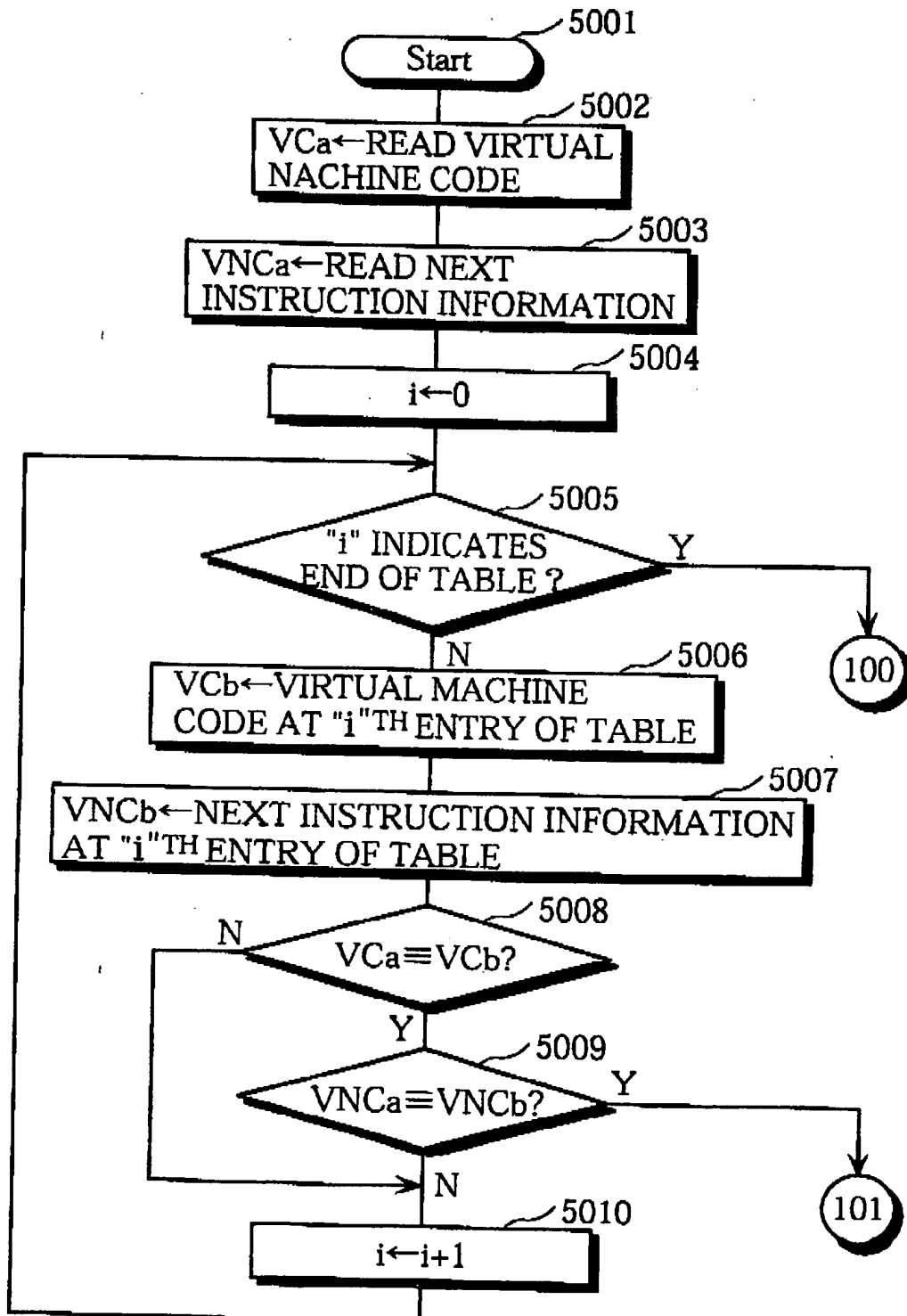


FIG. 45

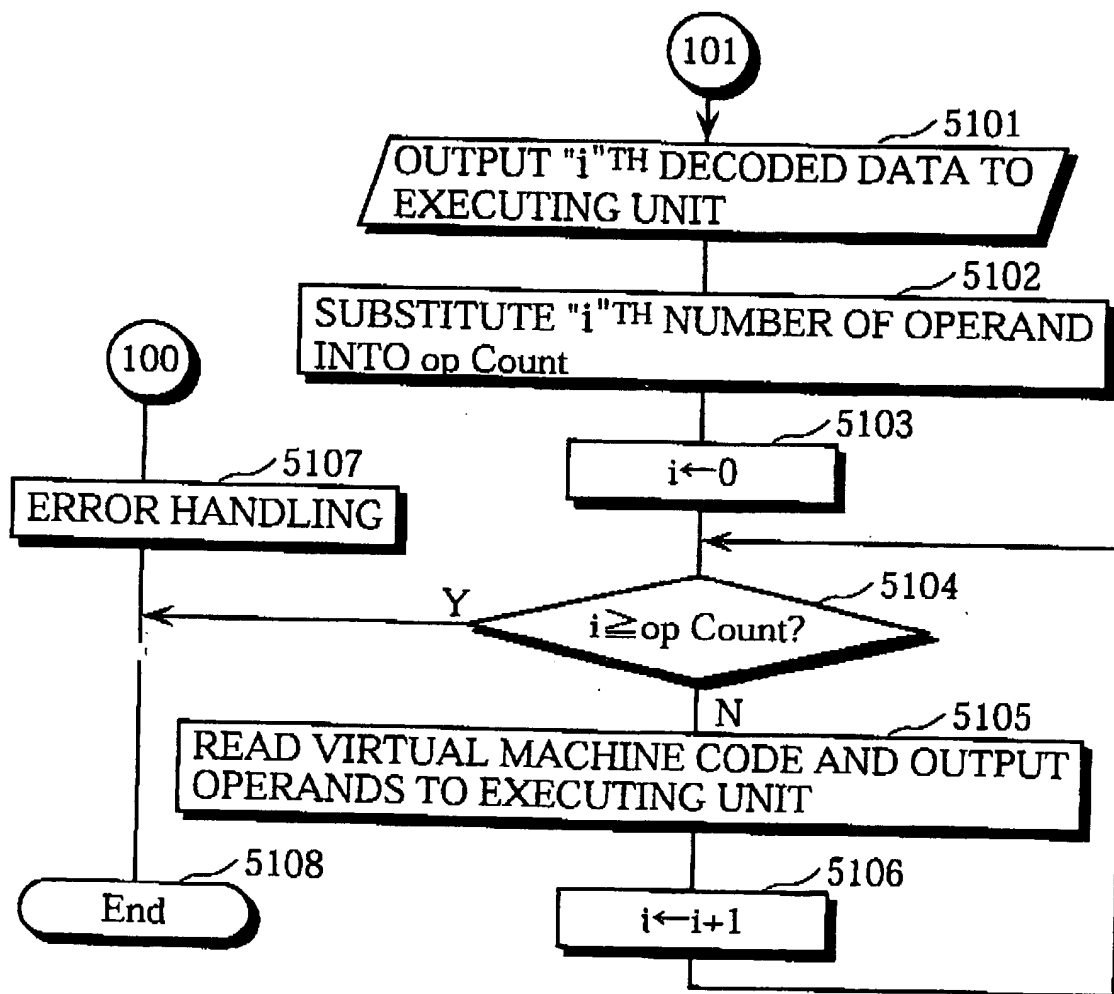


FIG. 46

1:<JUMP ADDRESS OF CODE TO PERFORM Push WITH "U">
2:OPERAND "2"
3:<JUMP ADDRESS OF CODE TO PERFORM Push WITH "U">
4:OPERAND "3"
5:<JUMP ADDRESS OF CODE TO PERFORM Push WITH "D">
6:OPERAND "4"
7:<JUMP ADDRESS OF CODE TO PERFORM Add WITH "D">
8:<JUMP ADDRESS OF CODE TO PERFORM Mult WITH "D">
9:<JUMP ADDRESS OF CODE TO PERFORM Pop WITH "U">
10:OPERAND "0"

FIG. 47A

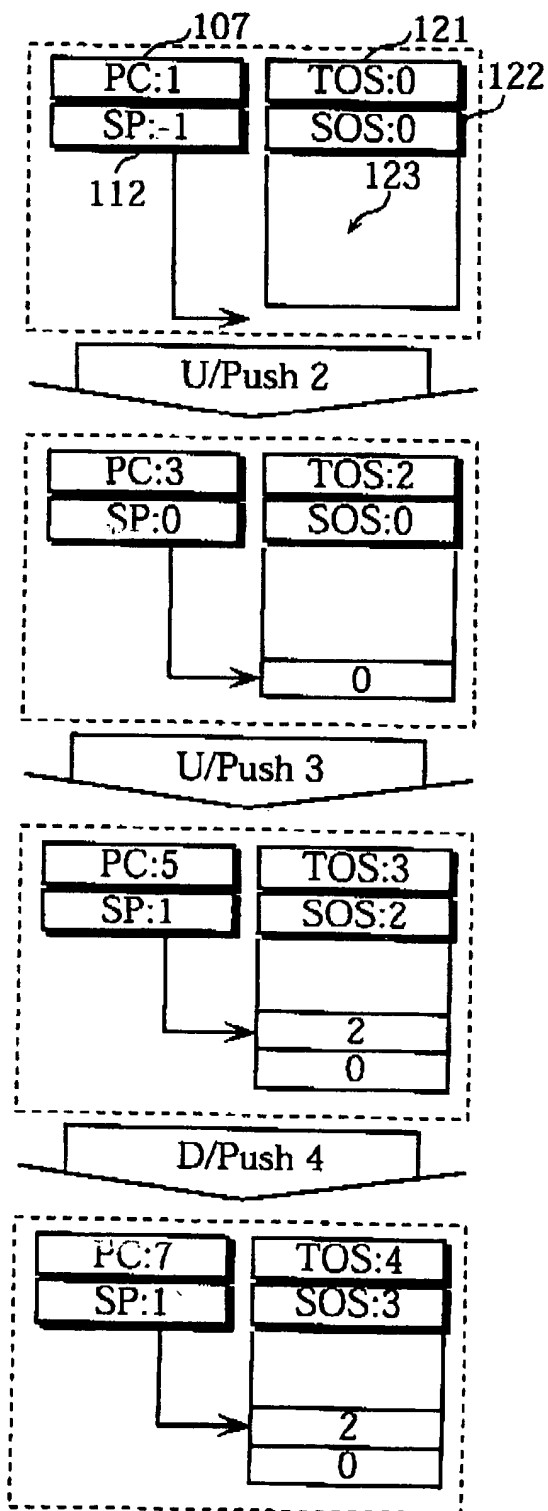


FIG. 47B

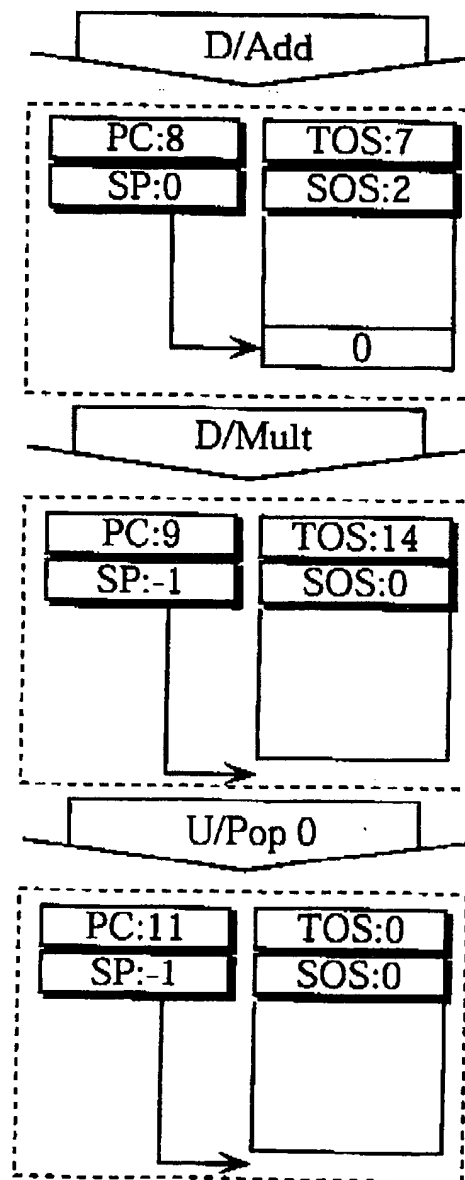




FIG. 48

CLOCK	1	2	3	4	5	6	7	8	9	10	11
Load r1,[r2]	IF	RF	ALU	MEM	WB						
Load r4,[r3]		IF	RF	ALU	MEM	WB					
Inc r2			IF	RF	ALU	MEM	WB				
Dec r3				IF	RF	ALU	MEM	WB			
Jmp r1					IF	RF	ALU	MEM	WB		
<hr/>											
Mult r0,r0,r4						IF	x				
							IF	RF	ALU	MEM	WB

FIG. 49

CLOCK	1	2	3	4	5	6	7	8	9
Load r1,[r2]	IF	RF	ALU	MEM	WB				
Load r4,[r3]	IF	RF	ALU	MEM	WB				
Inc r2		IF	RF	ALU	MEM	WB			
Dec r3		IF	RF	ALU	MEM	WB			
Jmp r1			IF	RF	ALU	MEM	WB		
<hr/>									
			IF	RF	x				
				IF	x				
				IF	x				
Mult r0,r0,r4					IF	RF	ALU	MEM	WB

FIG. 50

CLOCK	1	2	3	4	5	6	7	8	9	10	11
Load r1,[r2]	IF	RF	ALU	MEM	WB						
Load r4,[r3]		IF	RF	ALU	MEM	WB					
Inc r2			IF	RF	ALU	MEM	WB				
Dec r3				IF	RF	ALU	MEM	WB			
Jmp r1					IF	RF	ALU	MEM	WB		
						IF	x				
Mult r0,r0,r4							IF	RF	ALU	MEM	WB

FIG. 51

CLOCK	1	2	3	4	5	6	7	8	9	10
Load r1,[r2]	IF	RF	ALU	MEM	WB					
Load r4,[r3]	IF	RF	ALU	MEM	WB					
Inc r2		IF	RF	ALU	MEM	WB				
Dec r3		IF	RF	ALU	MEM	WB				
Jmp r1			IF	RF	.	ALU	MEM	WB		
			IF	RF	ALU	x				
				IF	RF	x				
				IF	RF	x				
					IF	x				
					IF	x				
Mult r0,r0,r4						IF	RF	ALU	MEM	WB

FIG. 52

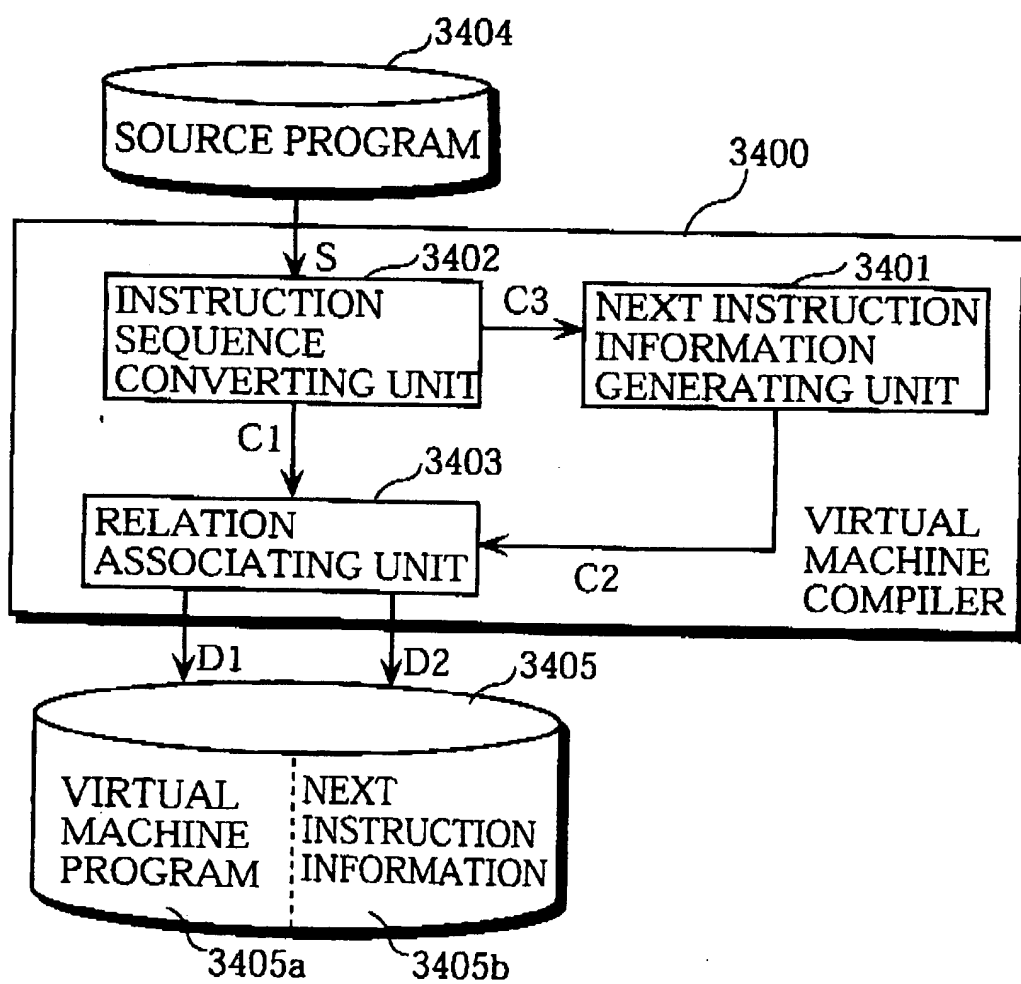




FIG. 54

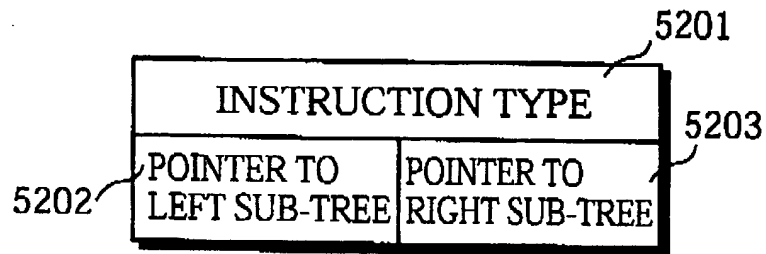


FIG. 55

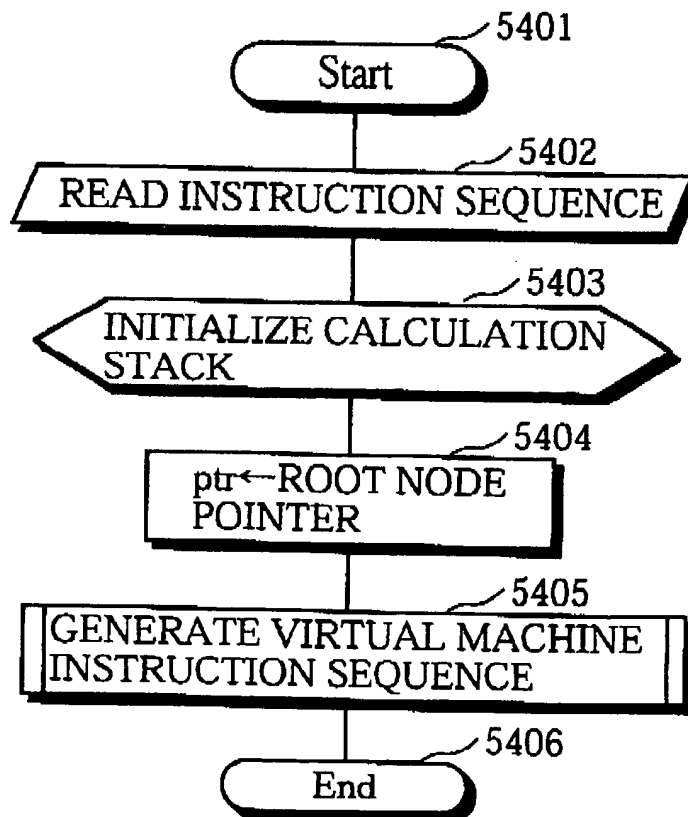
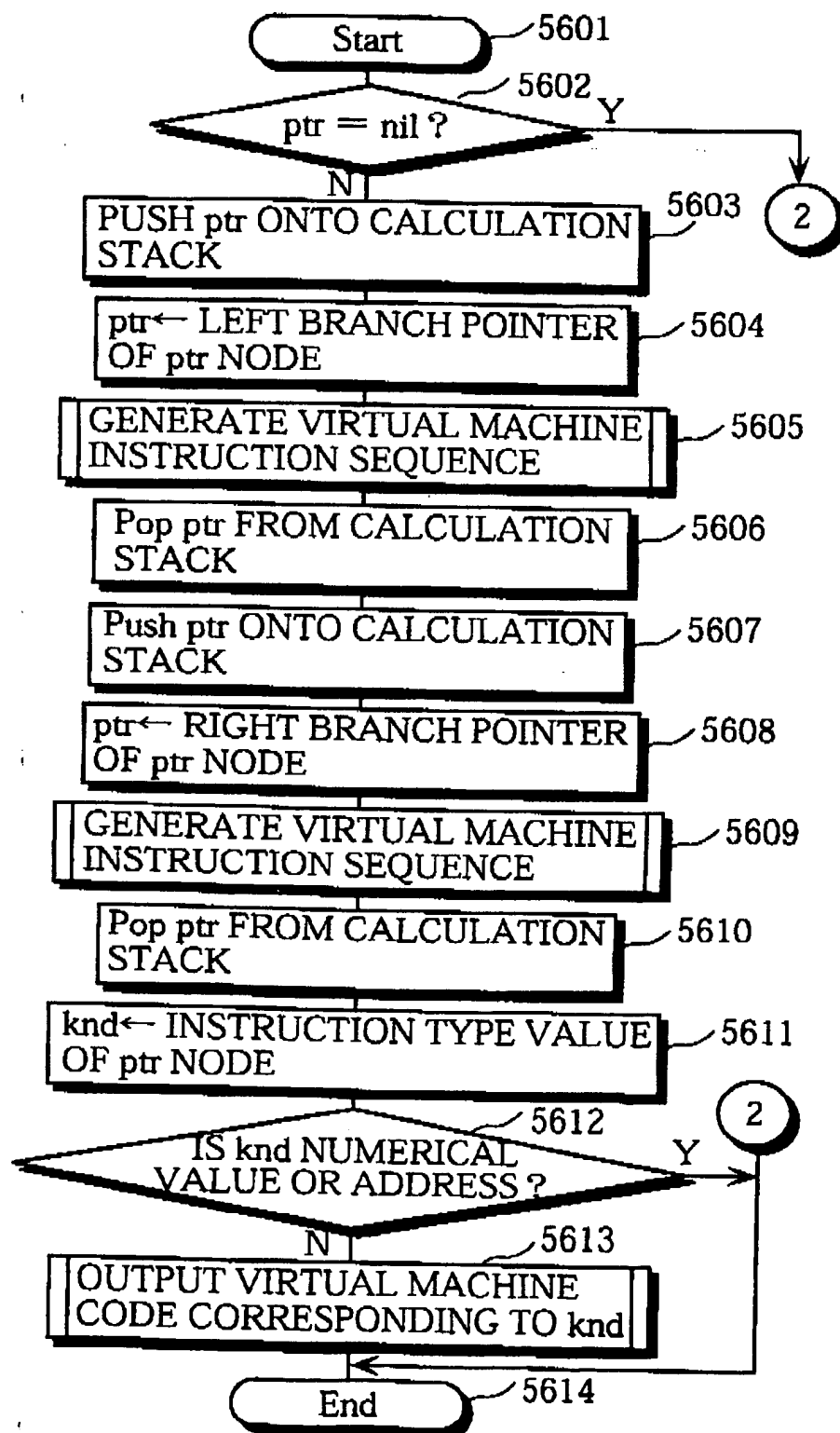


FIG. 56



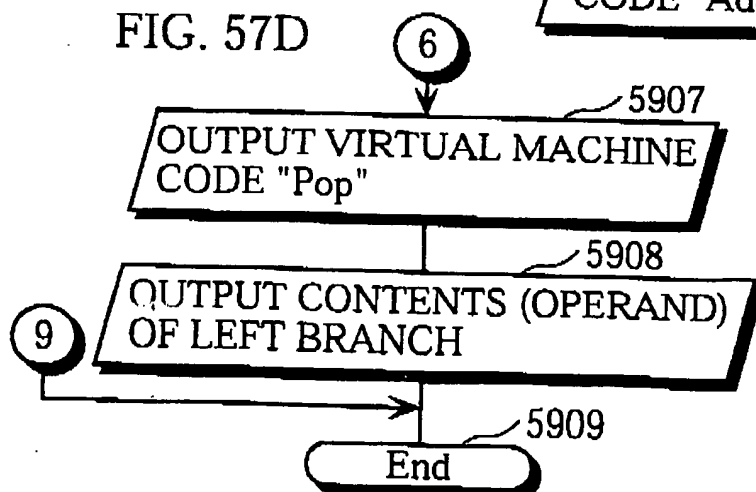
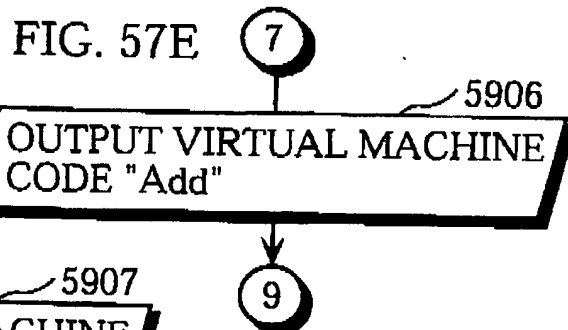
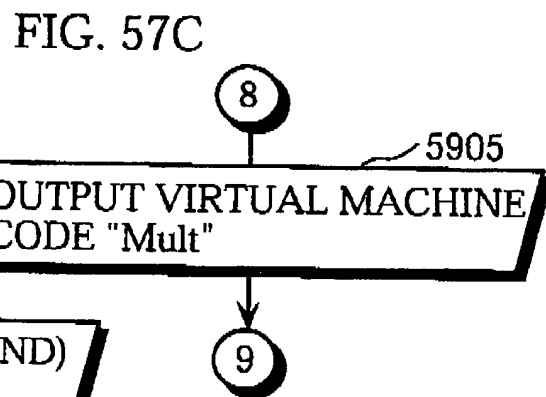
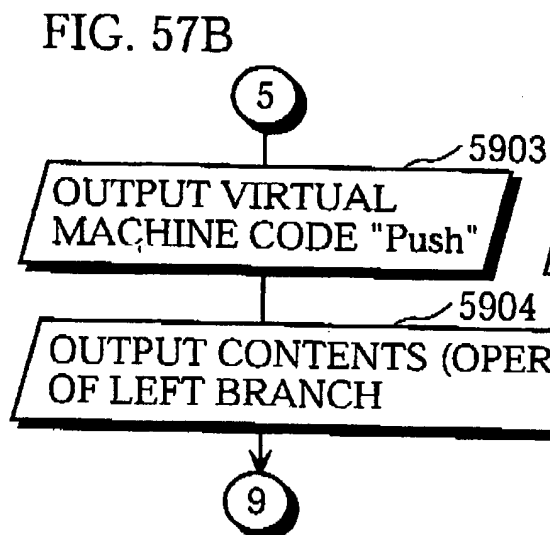
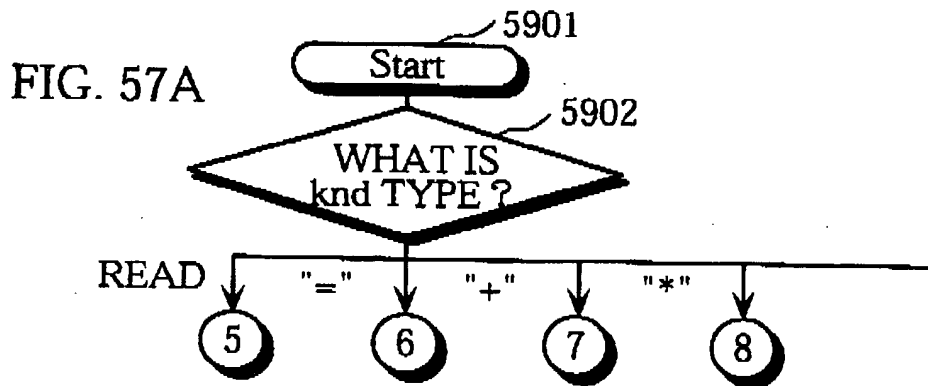






FIG. 59

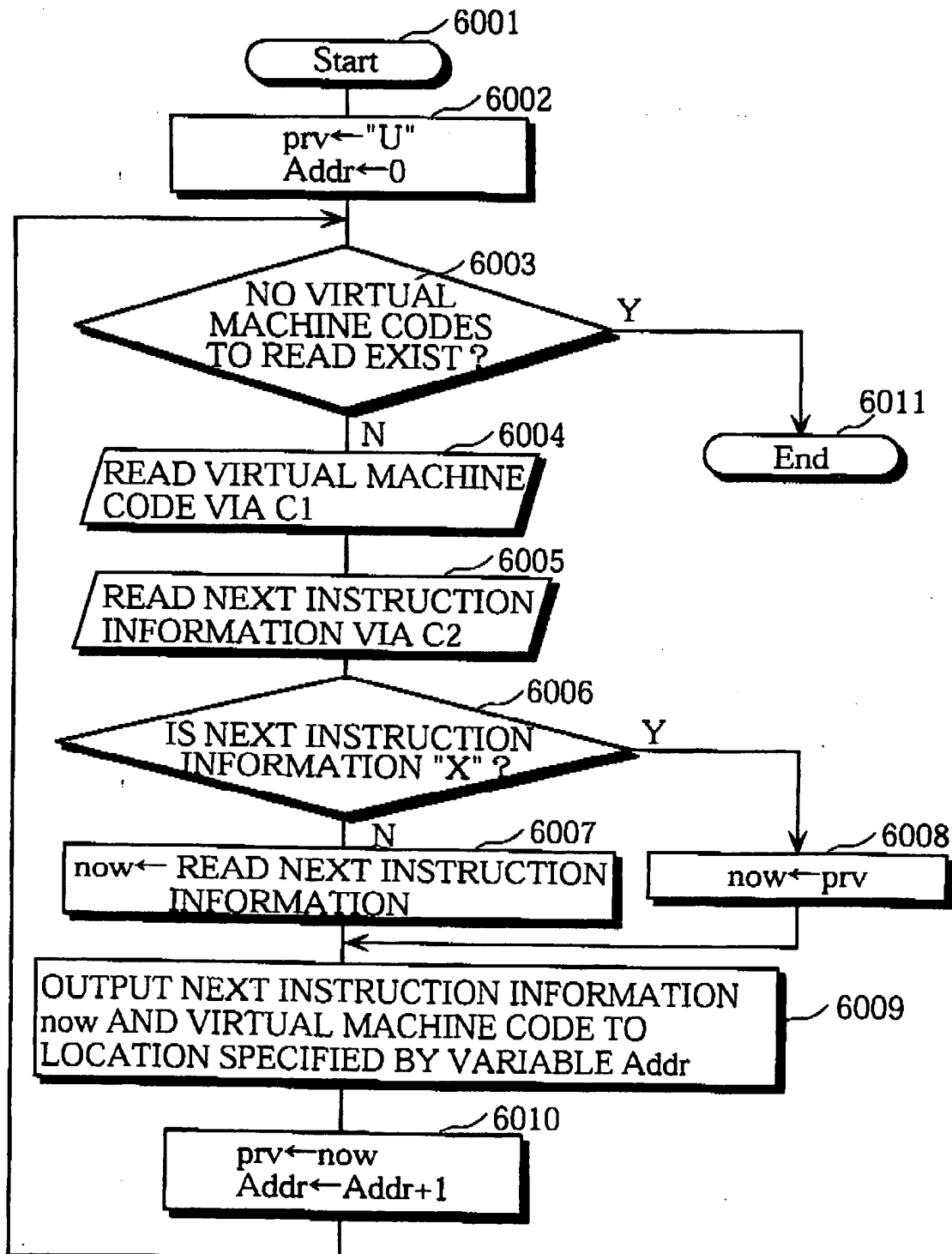


FIG. 60

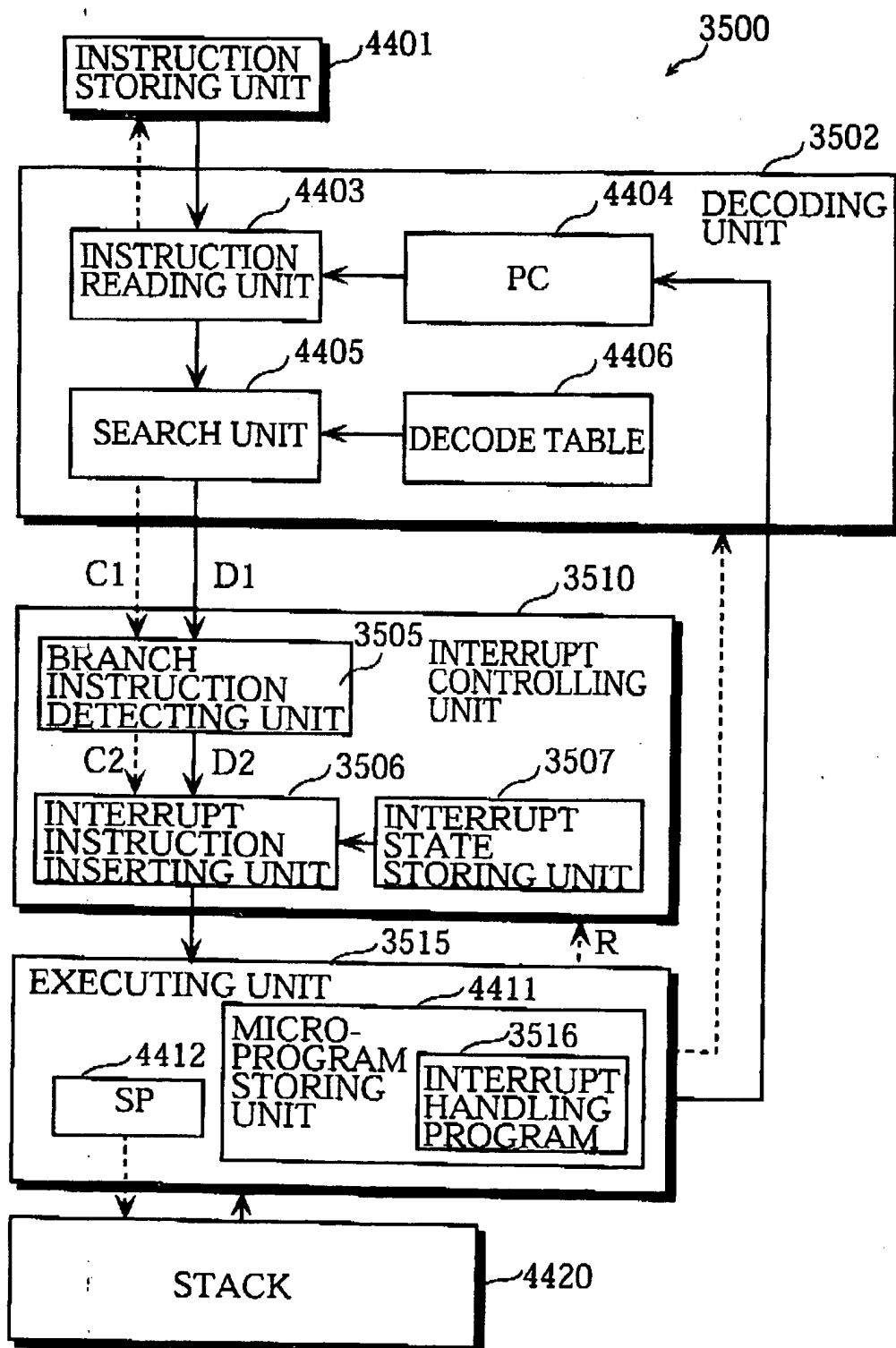


FIG. 61

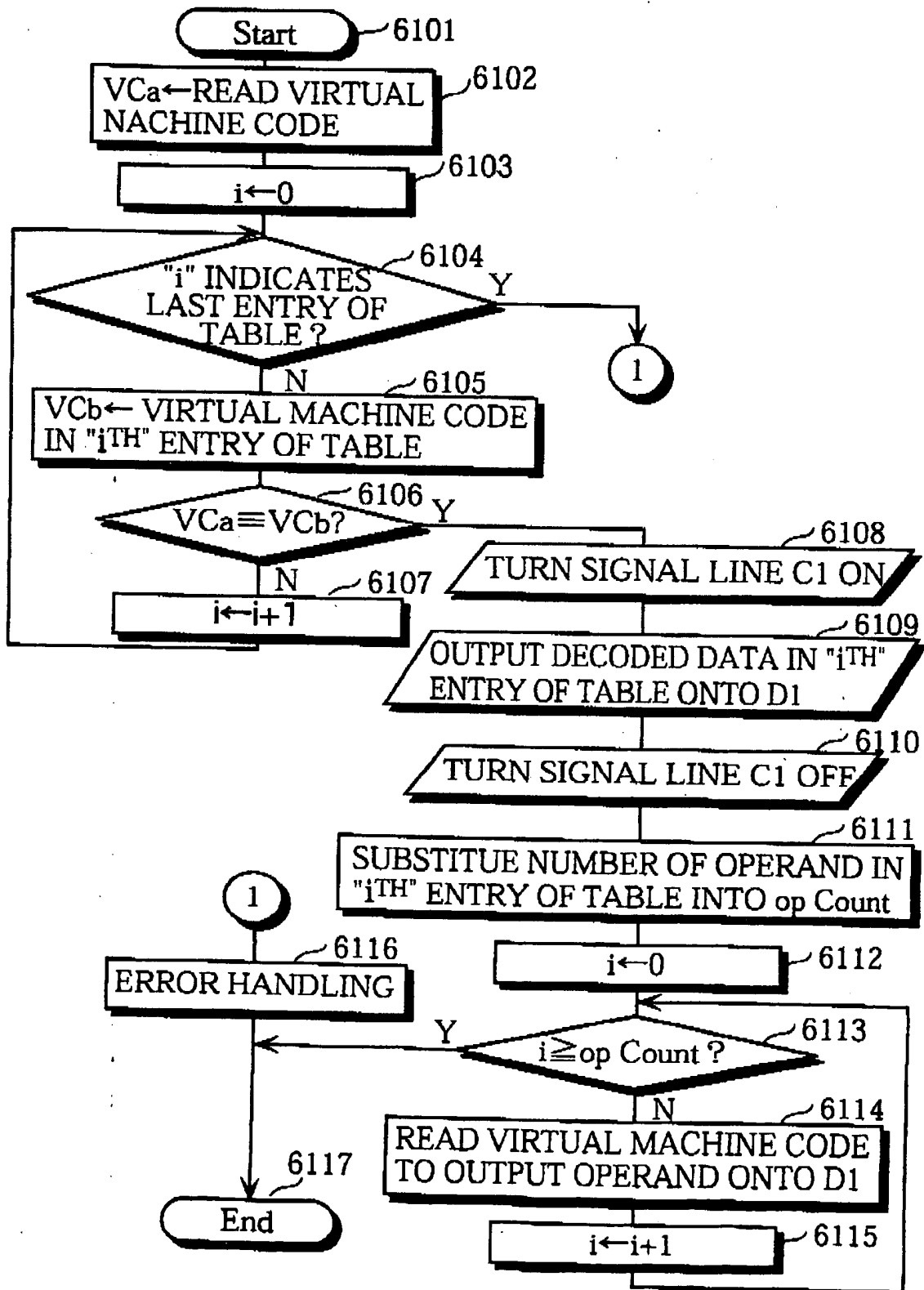


FIG. 62

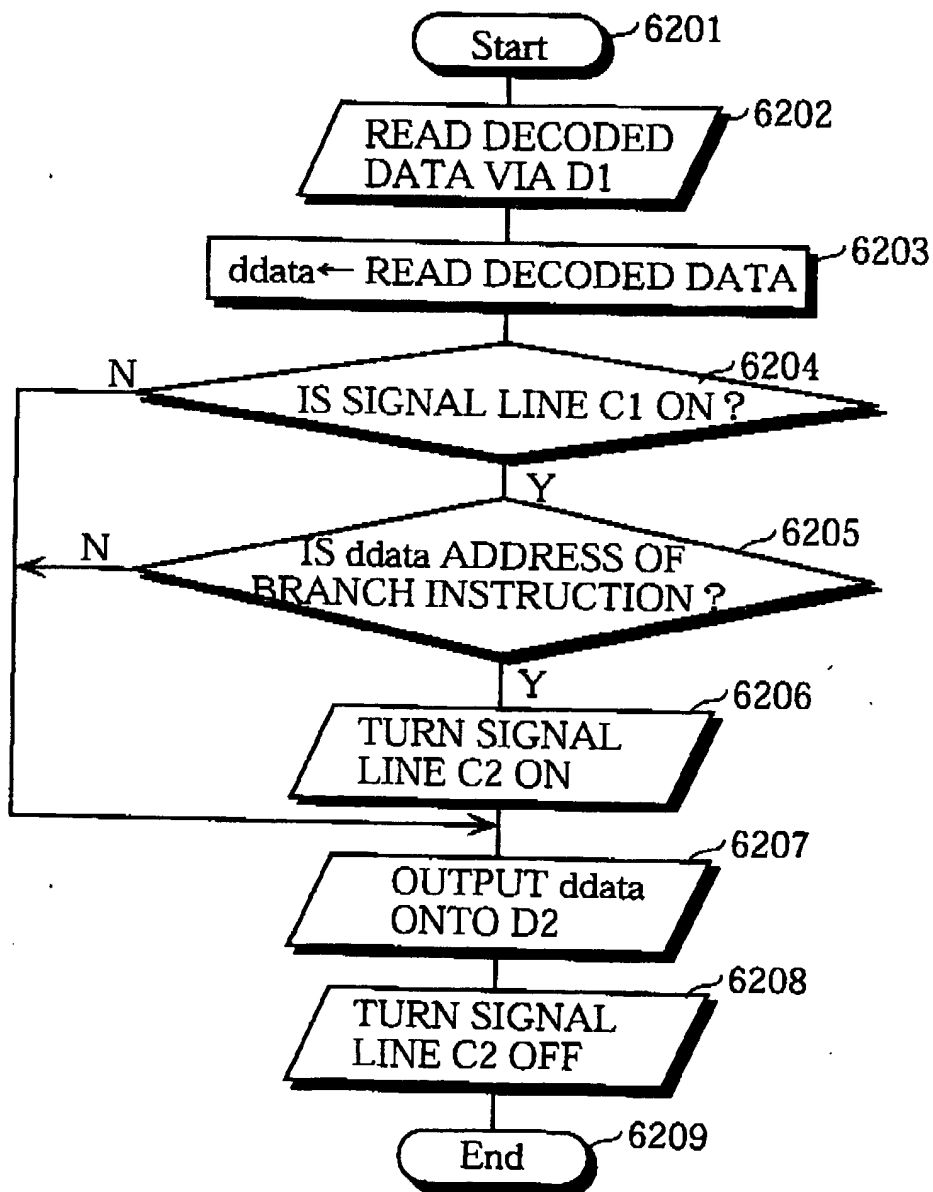


FIG. 63

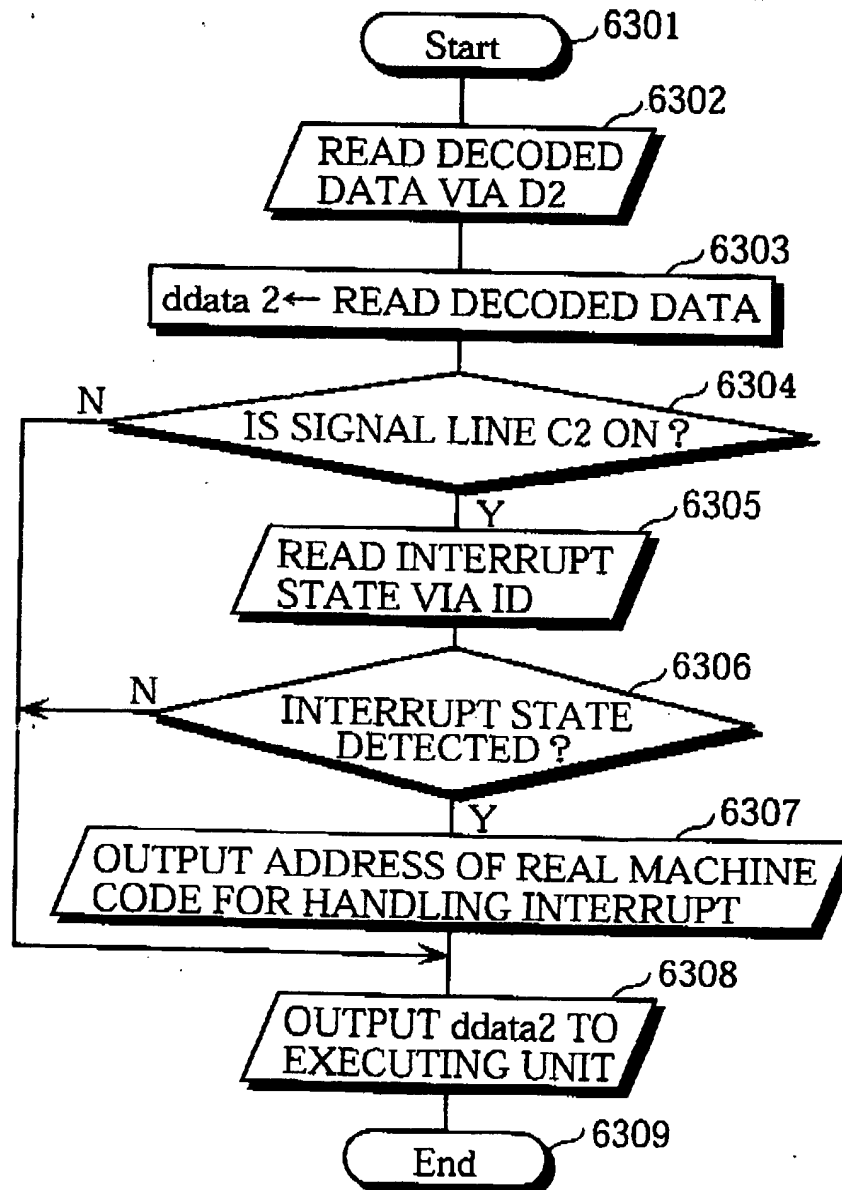


FIG 64

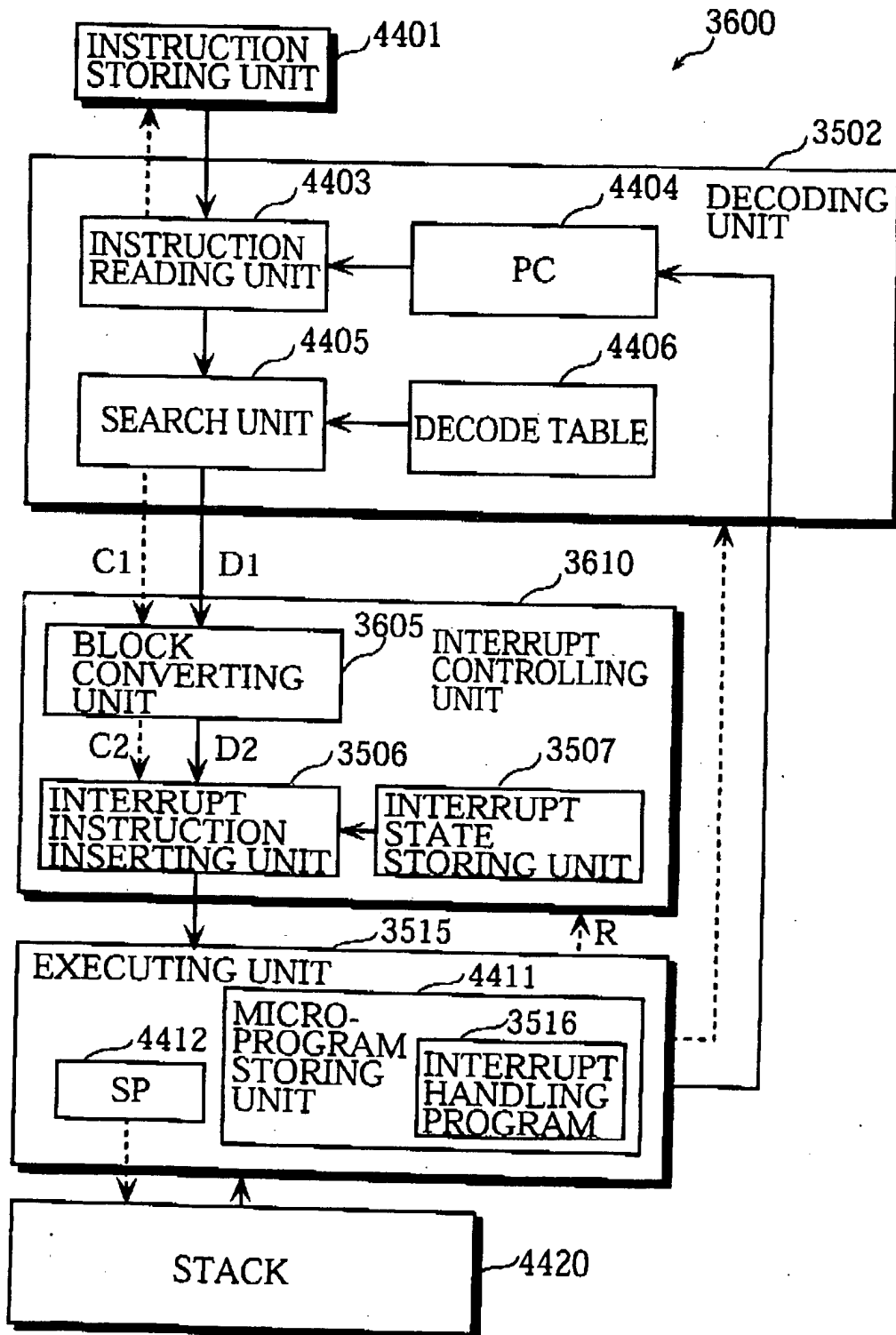


FIG. 65

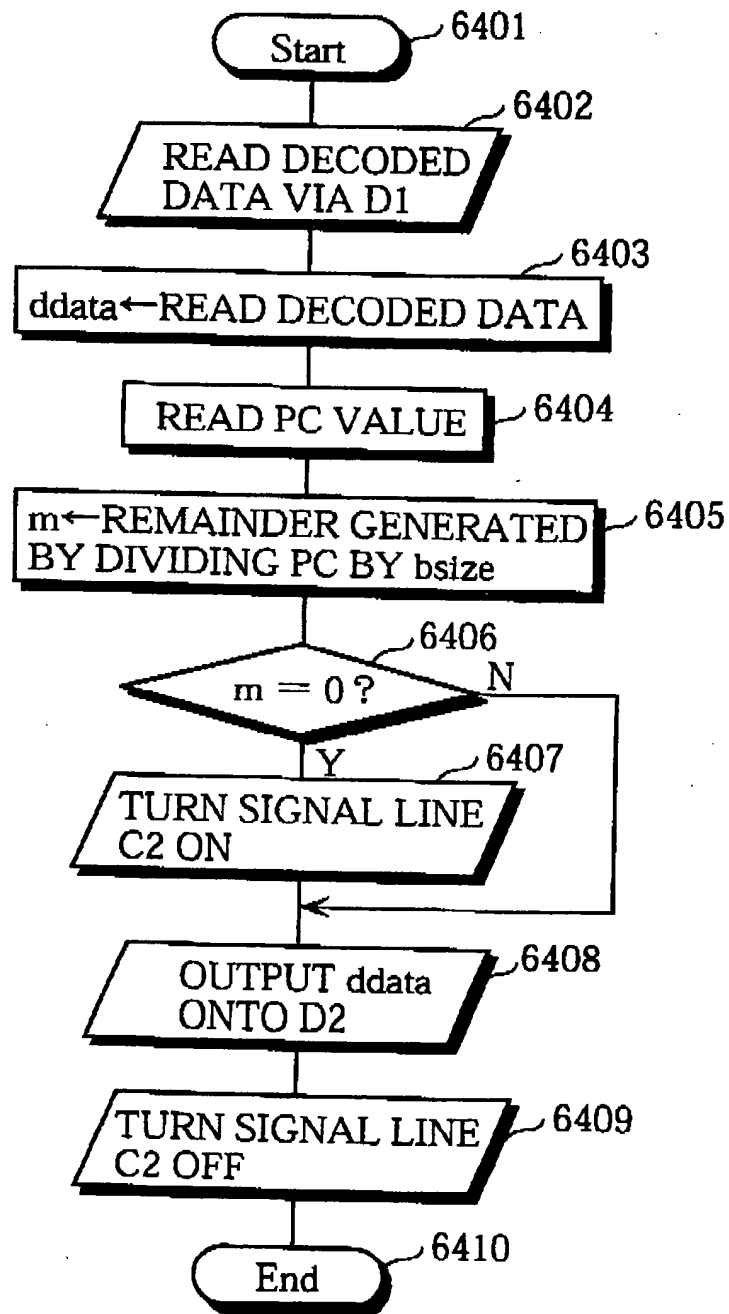


FIG. 66

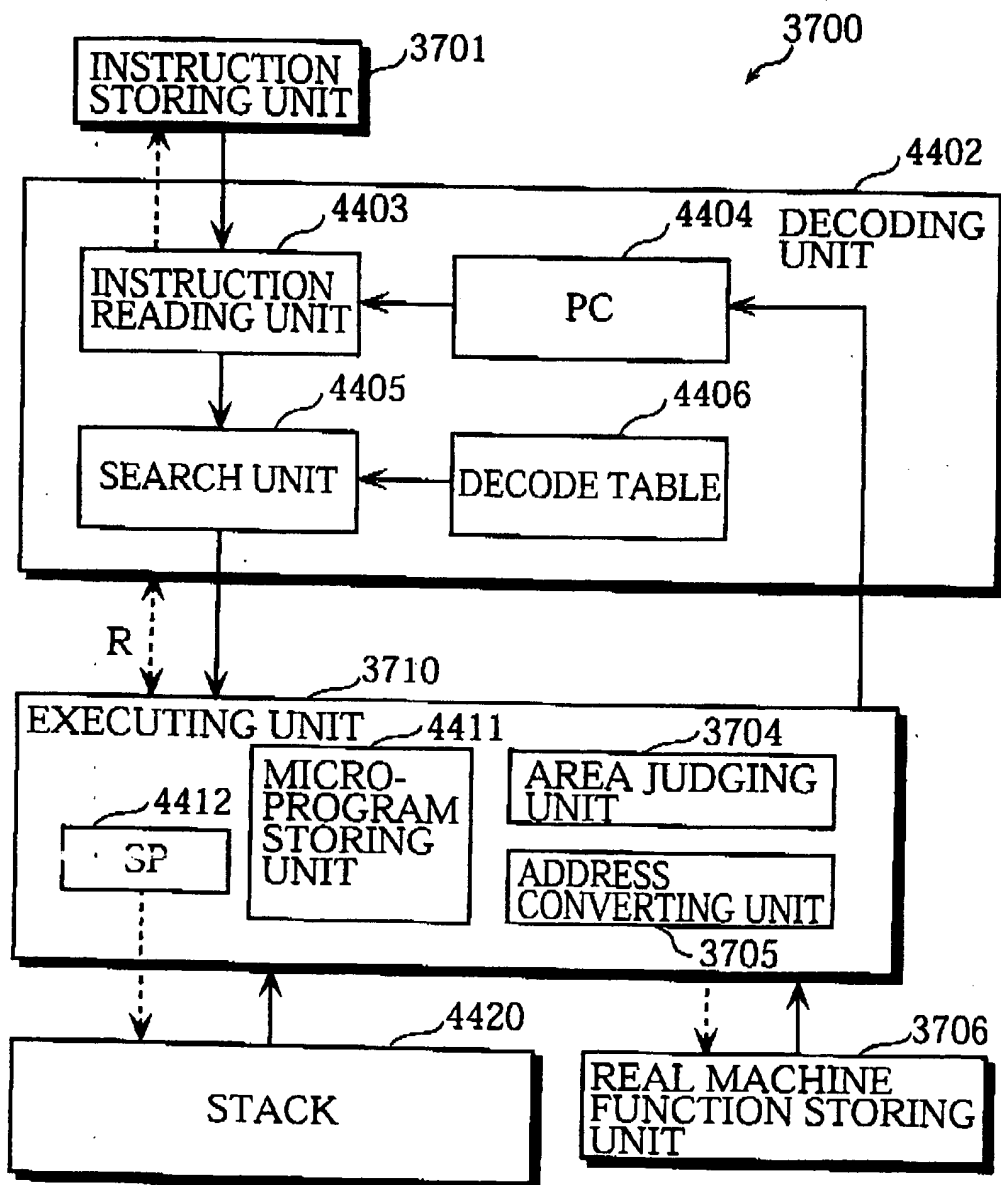




FIG. 67

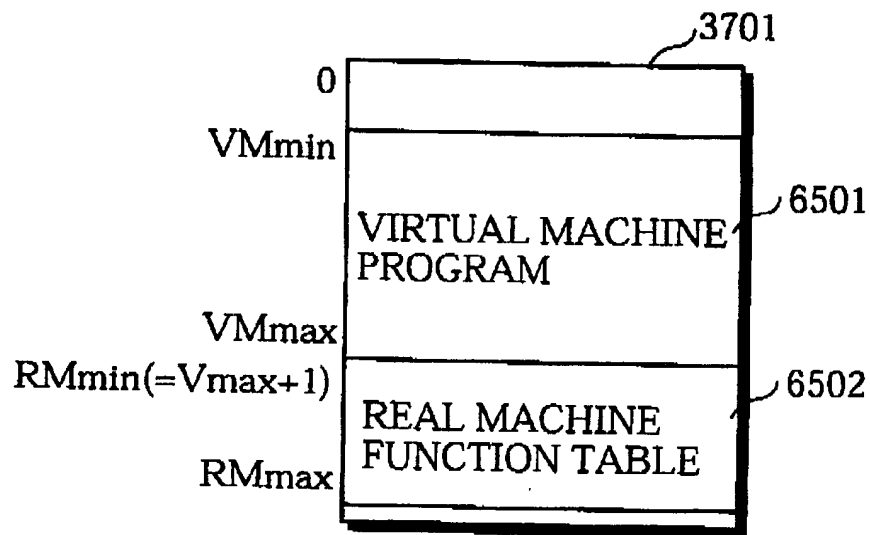


FIG. 68

(CORRESPONDING ADDRESS)

RMmax	(RMmax-RMmin) <sup>th</sup> POINTER TO REAL MACHINE FUNCTION
RMmax-1	(RMmax-RMmin-1) <sup>th</sup> POINTER TO REAL MACHINE FUNCTION
RMmax-2	(RMmax-RMmin-2) <sup>th</sup> POINTER TO REAL MACHINE FUNCTION
	:
RMmin	0 <sup>th</sup> POINTER TO REAL MACHINE FUNCTION

FIG. 69

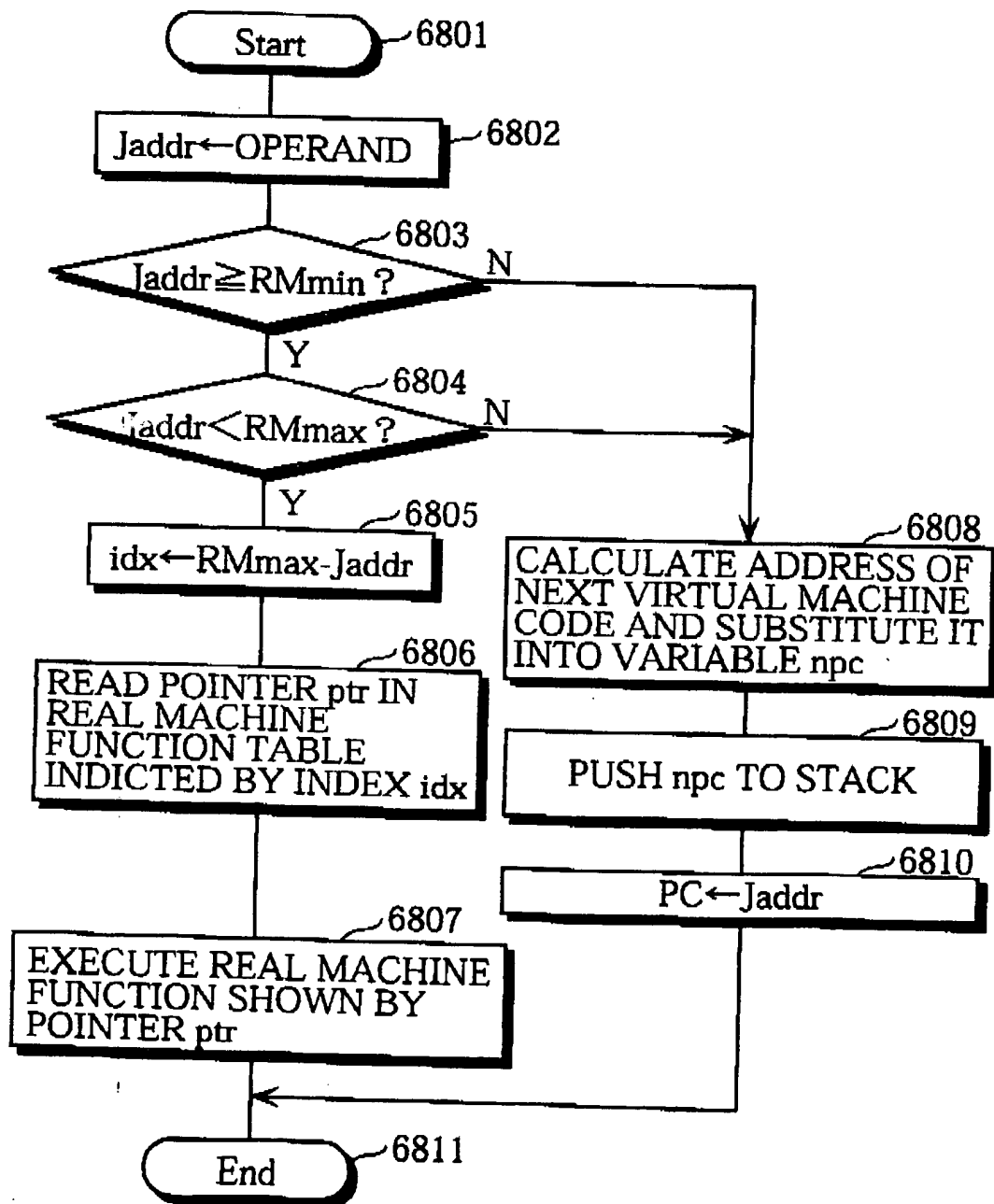




FIG. 71

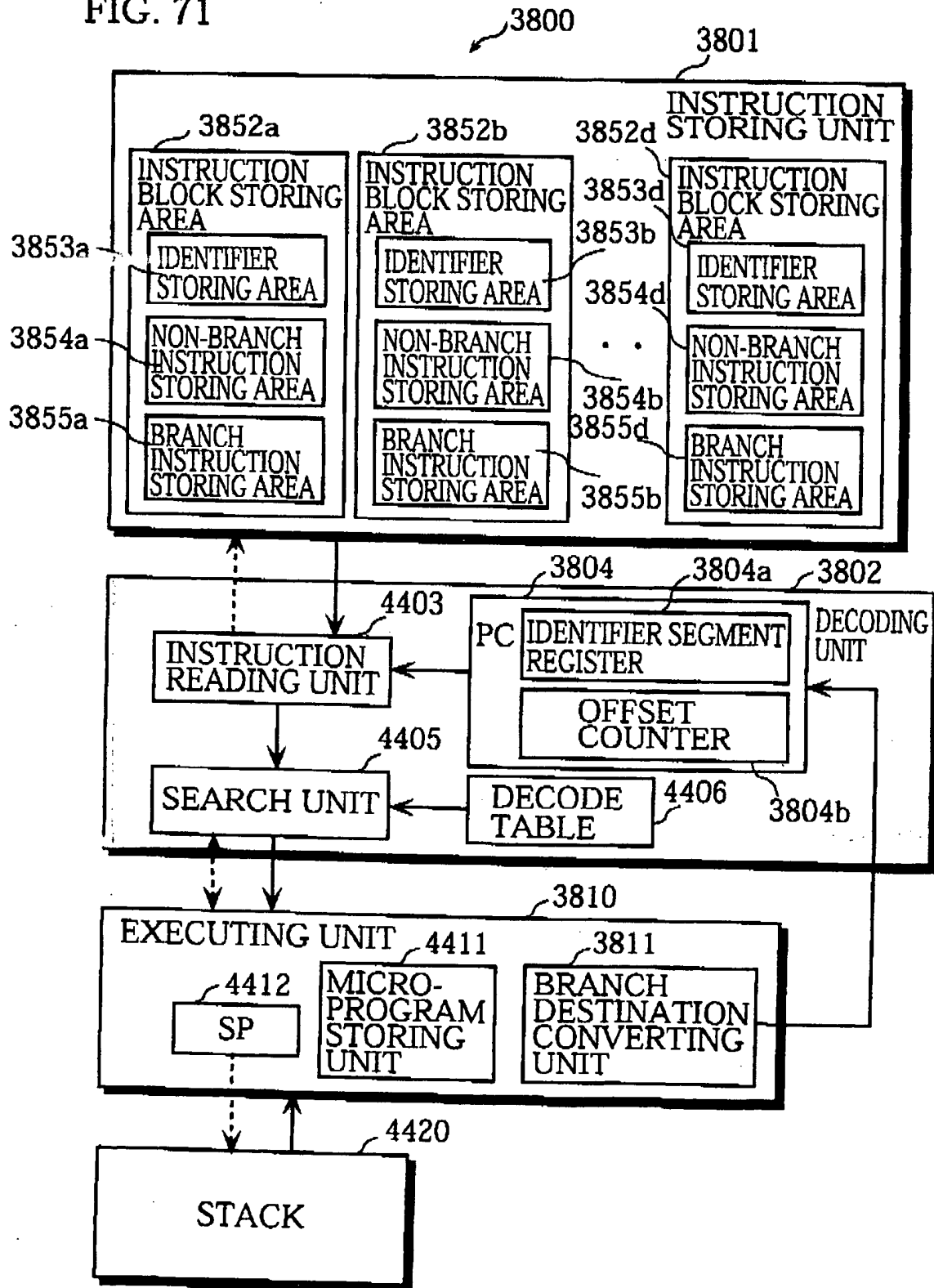


FIG. 72

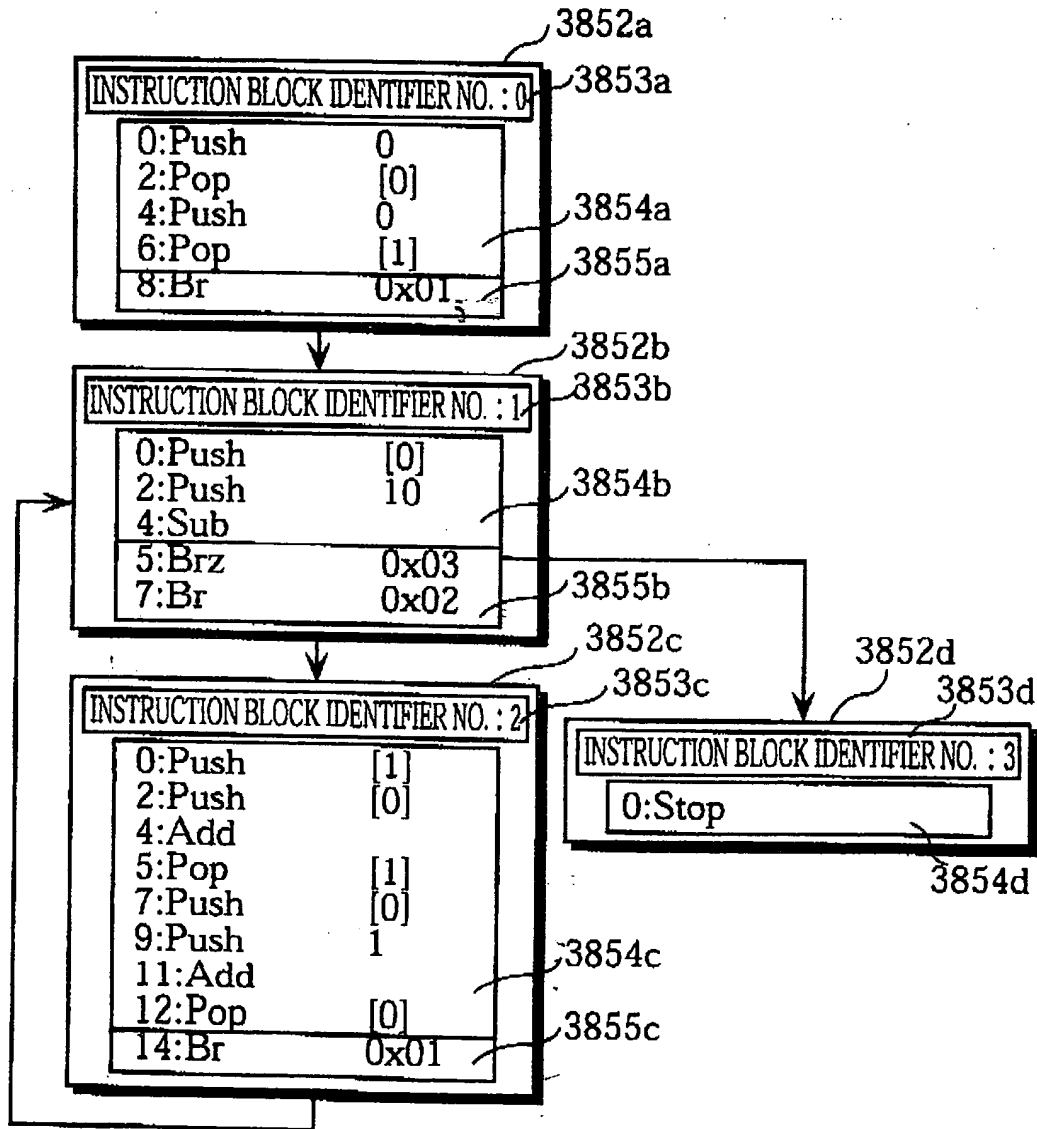


FIG. 73

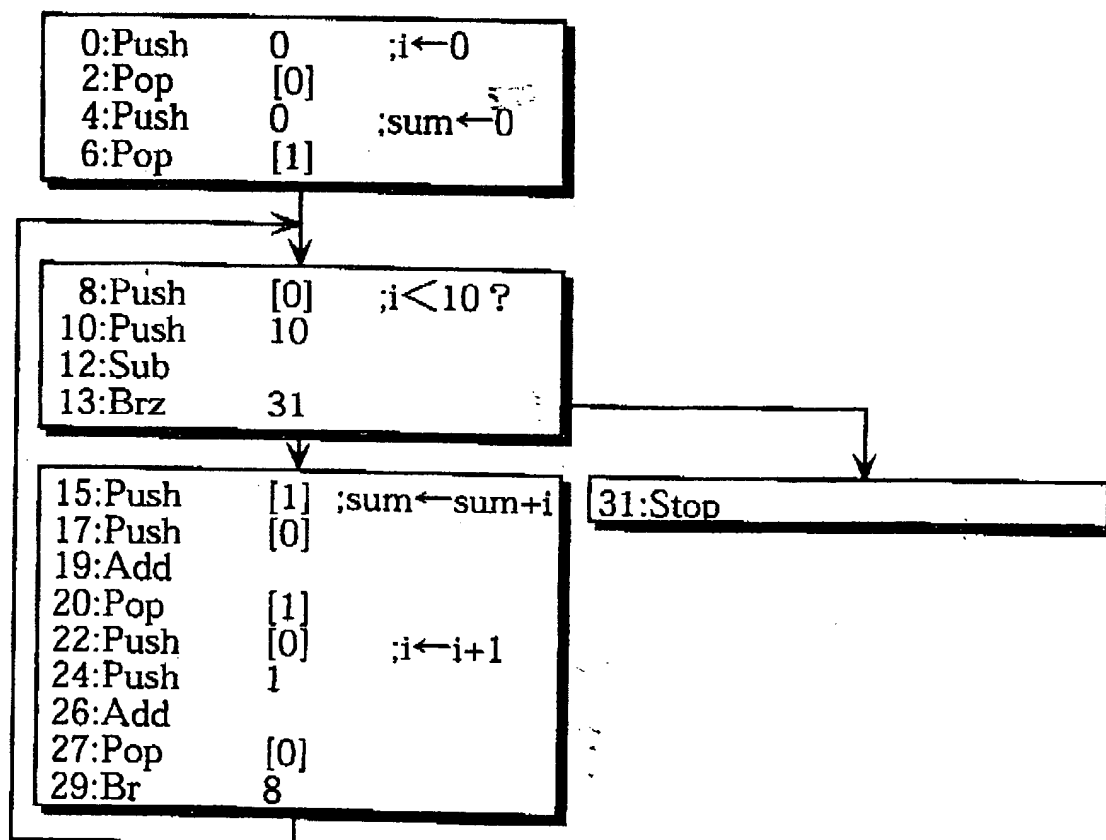


FIG. 74



FIG. 75

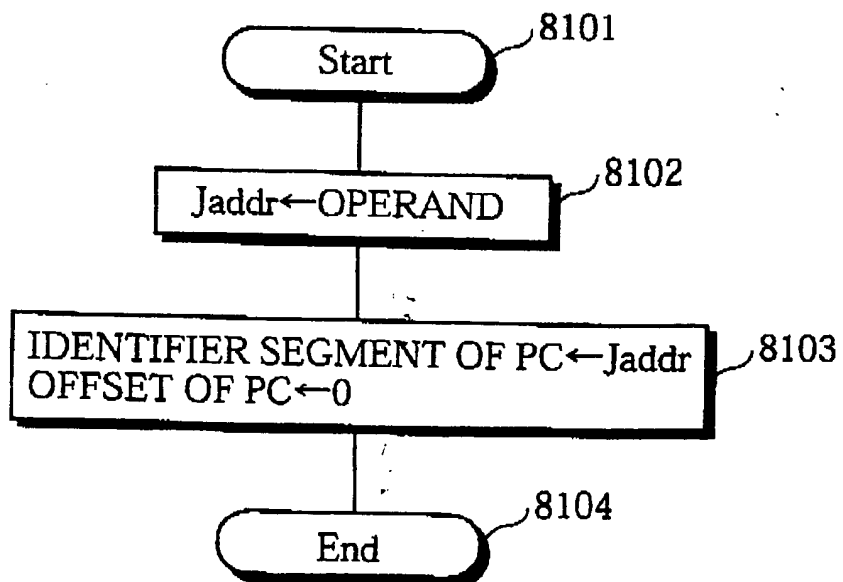


FIG. 76

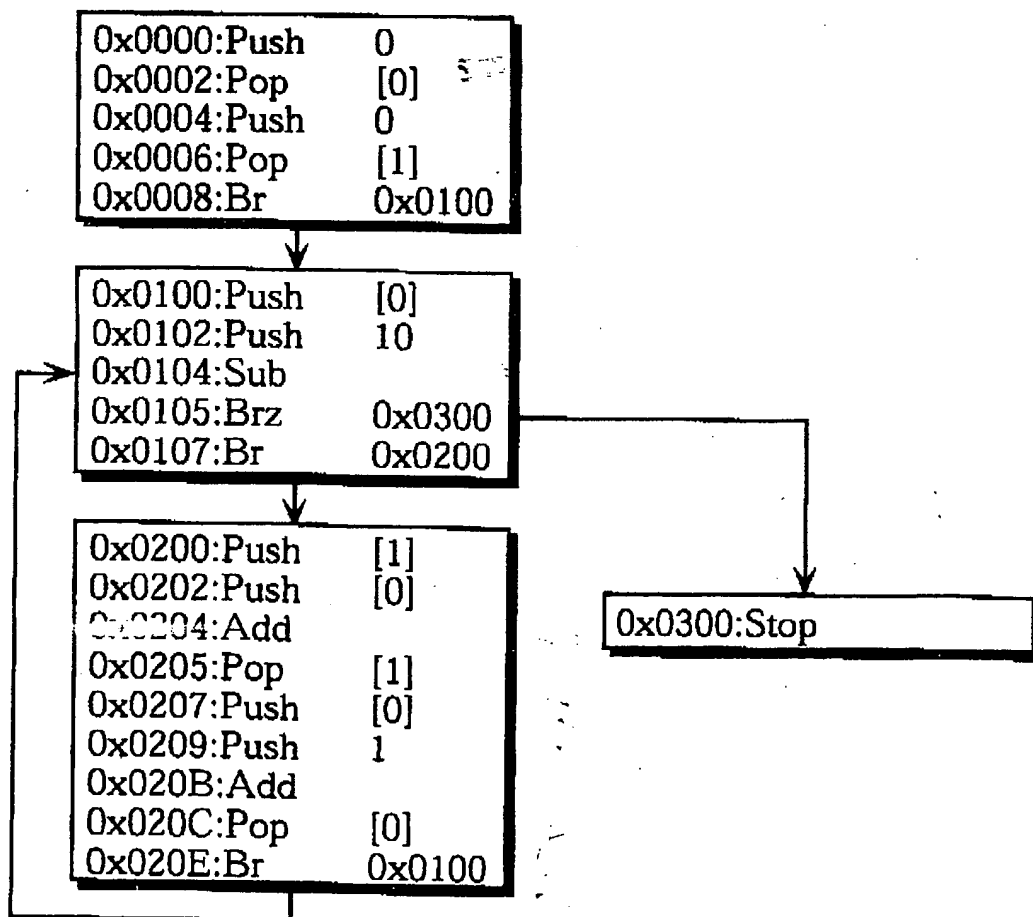




FIG. 77

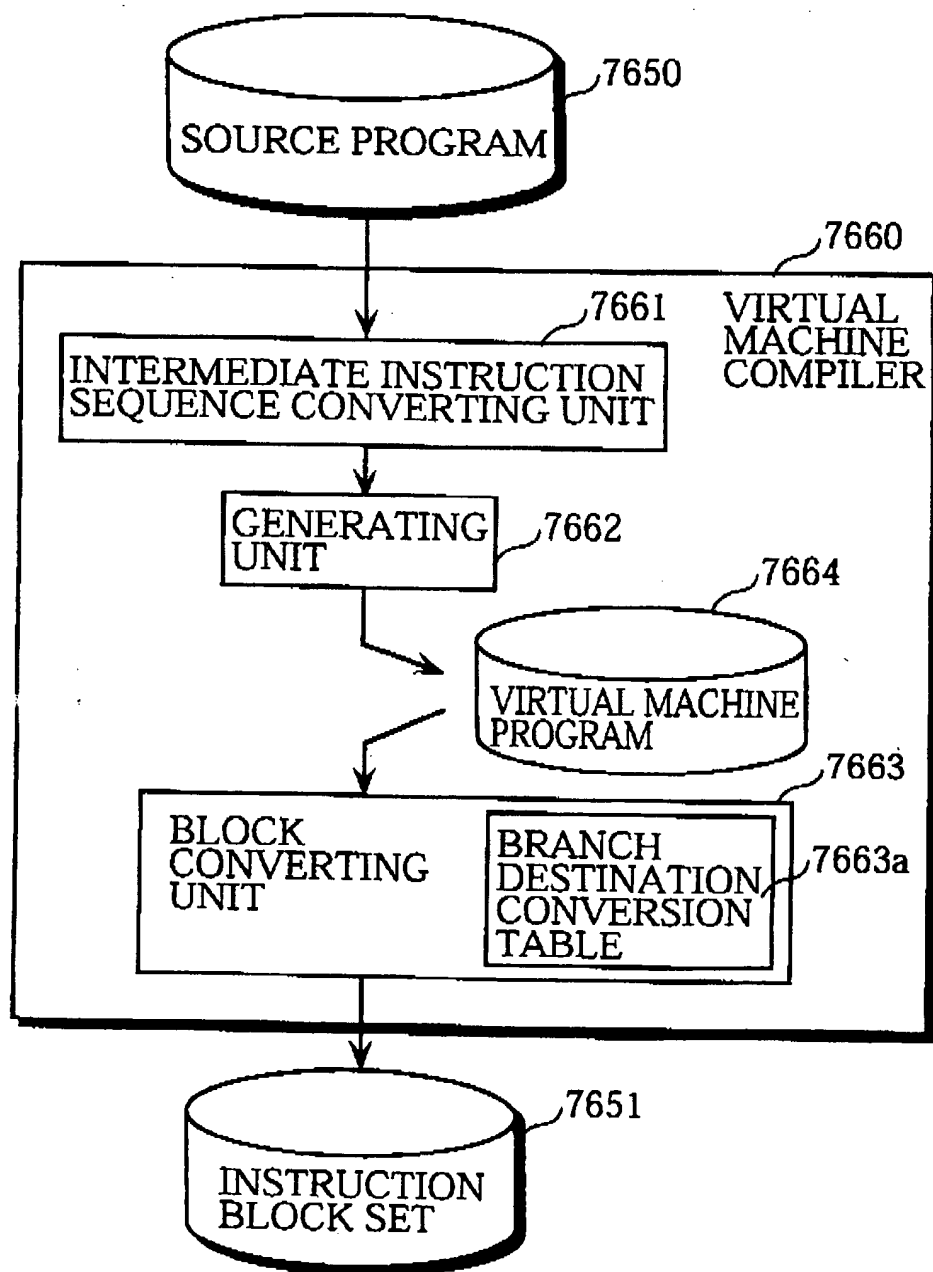




FIG. 79

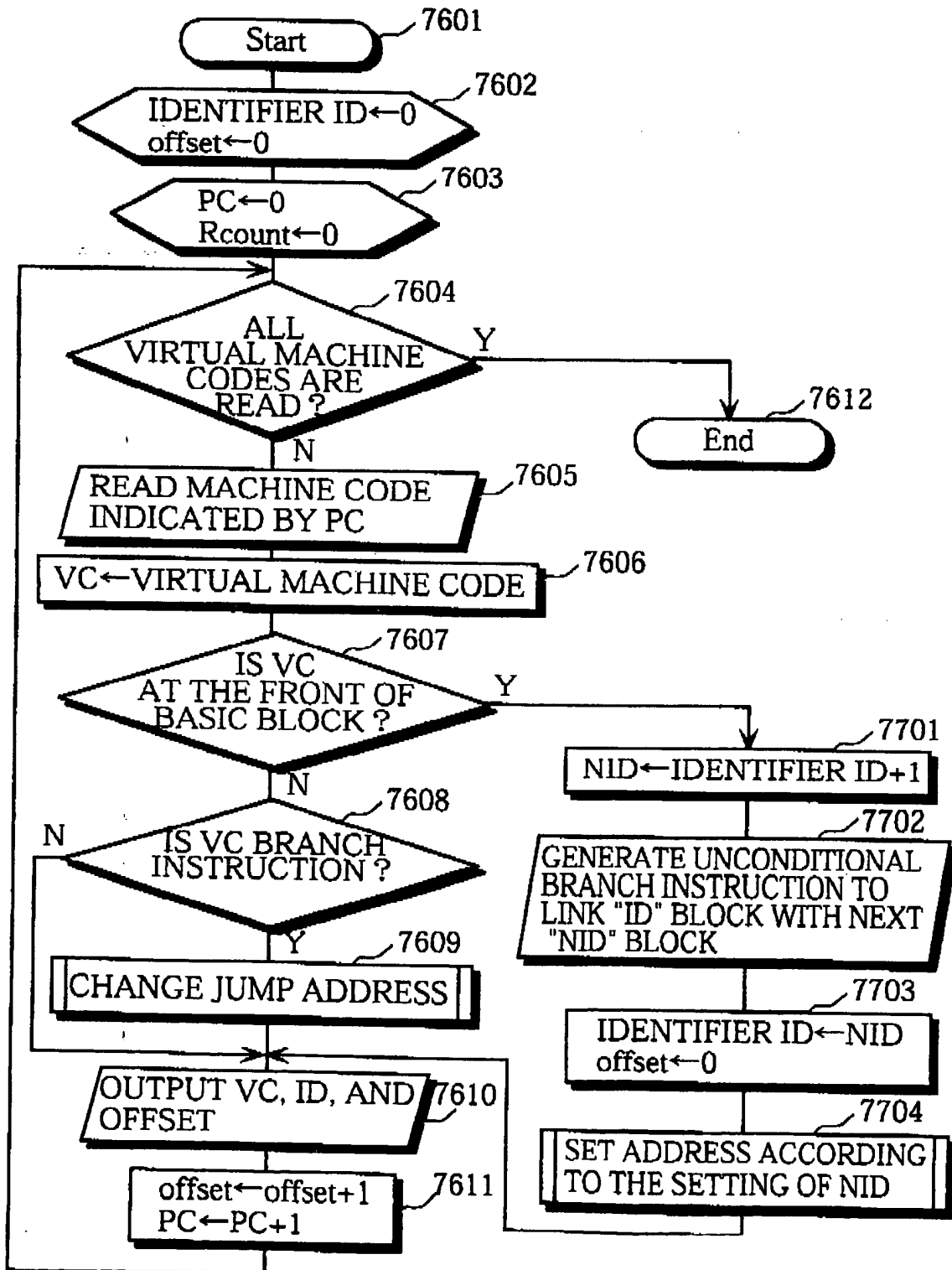


FIG. 80

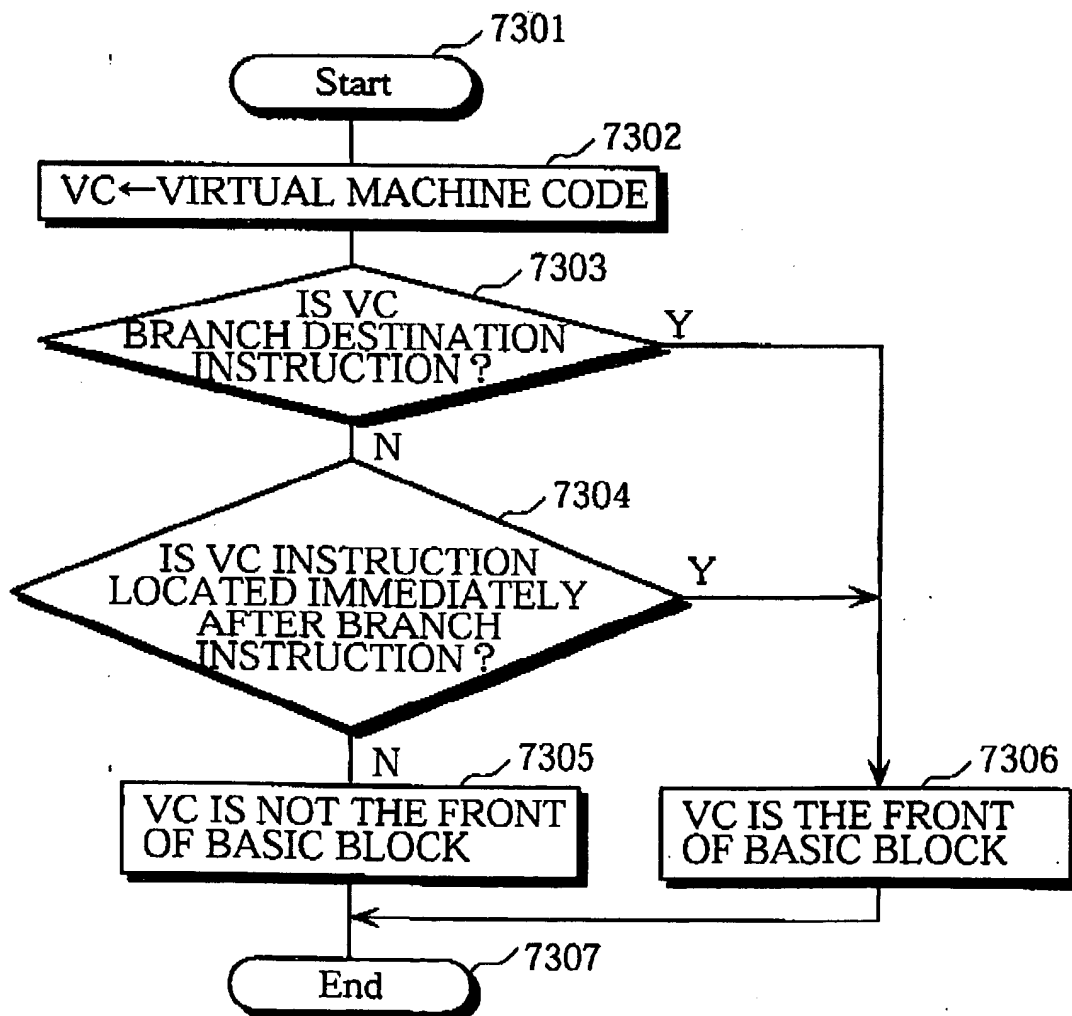


FIG. 81

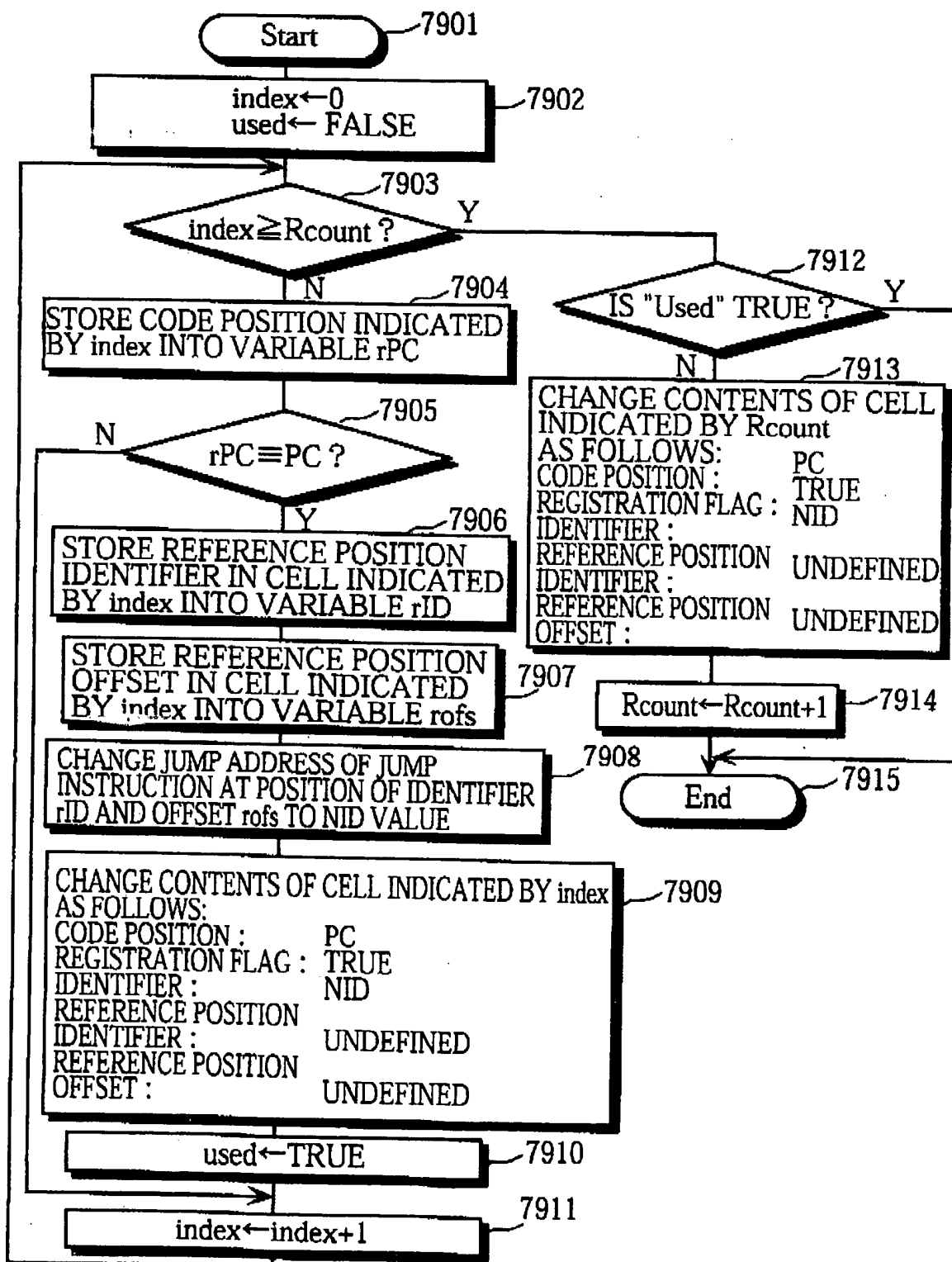


FIG. 82

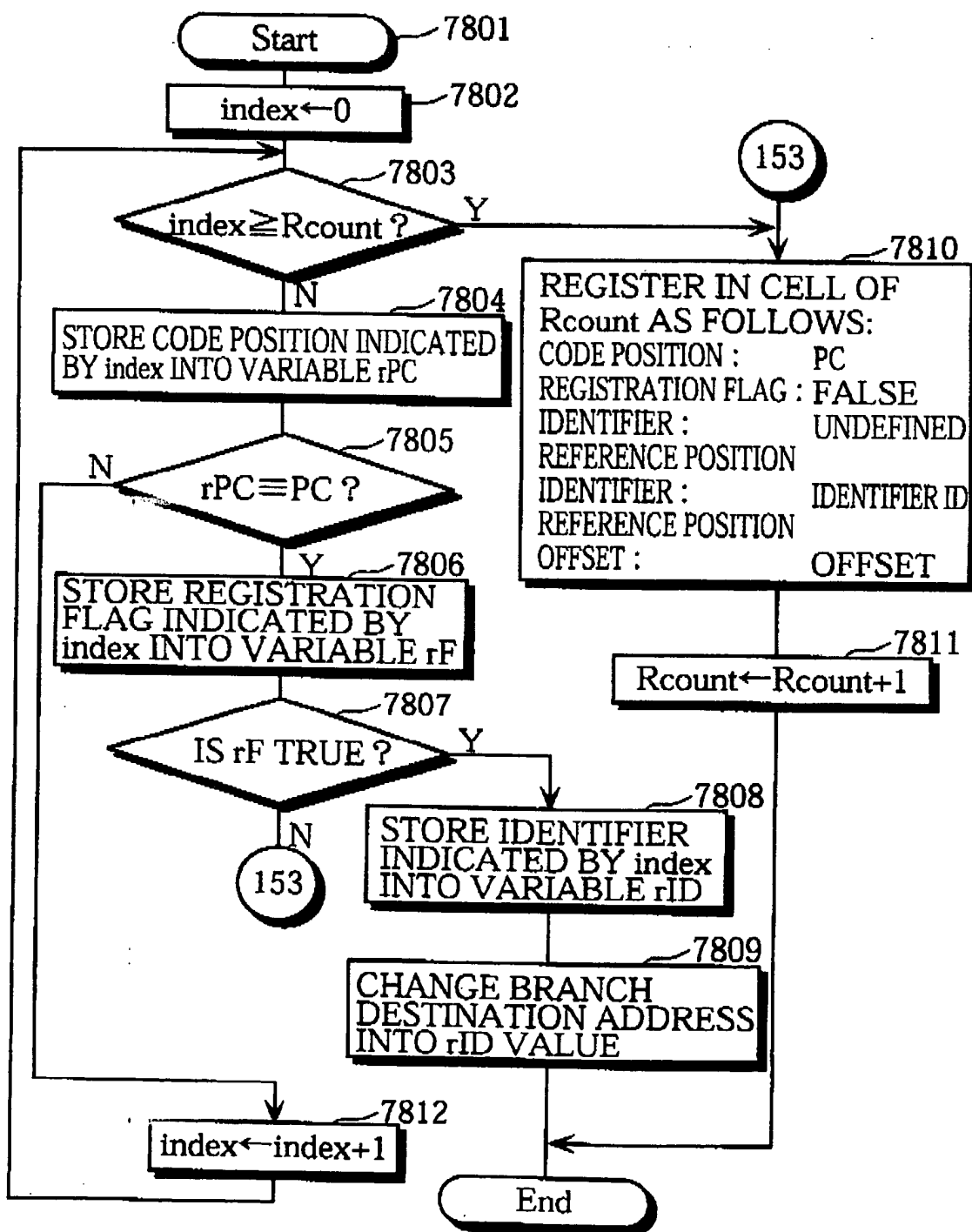


FIG. 83

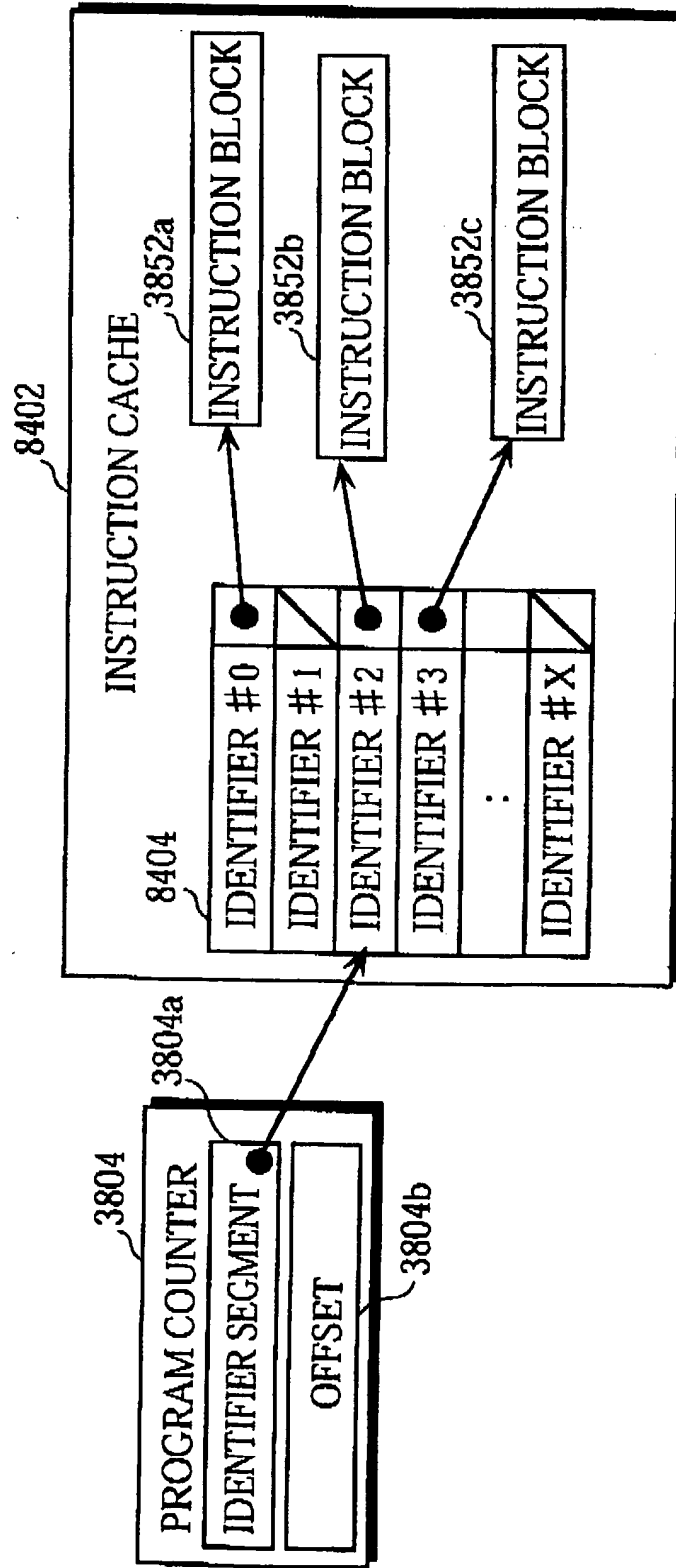


FIG. 84

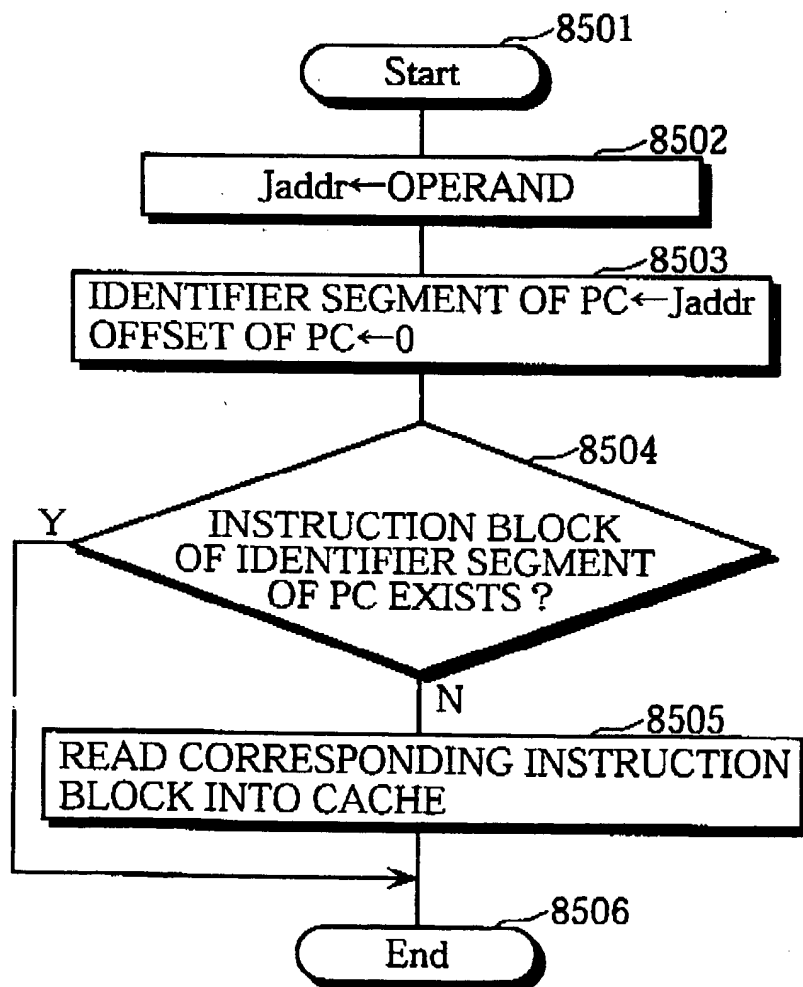




FIG. 85

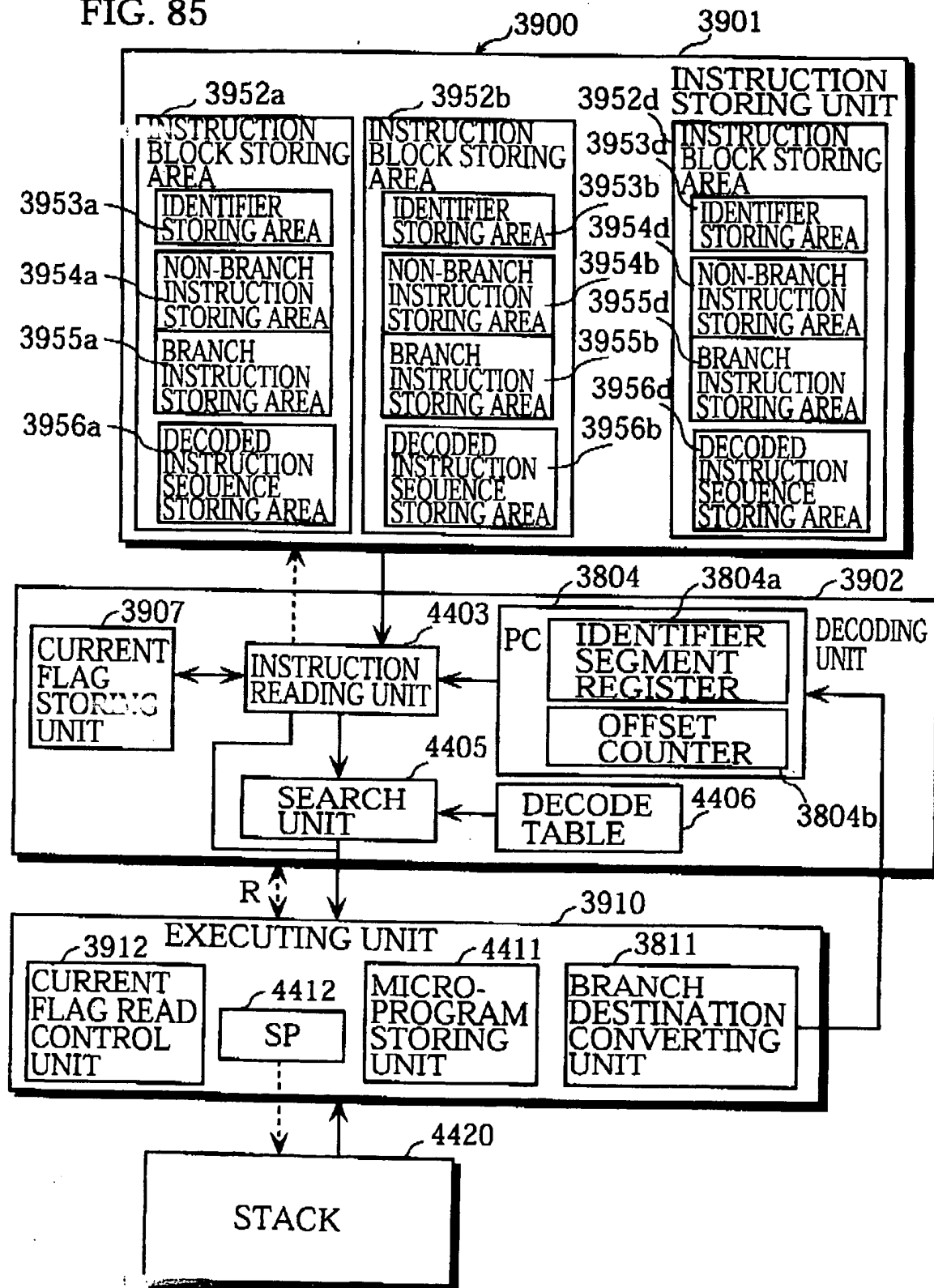


FIG. 86A

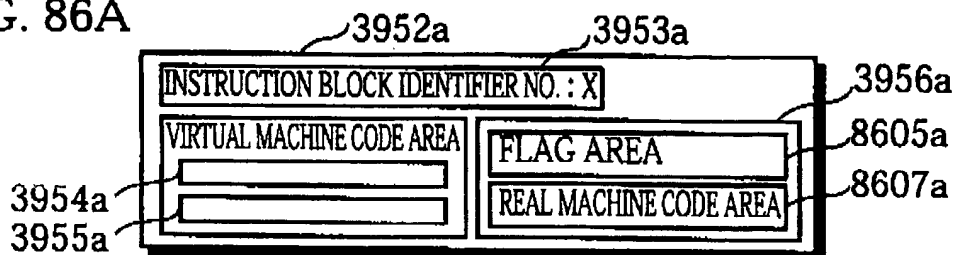


FIG. 86B

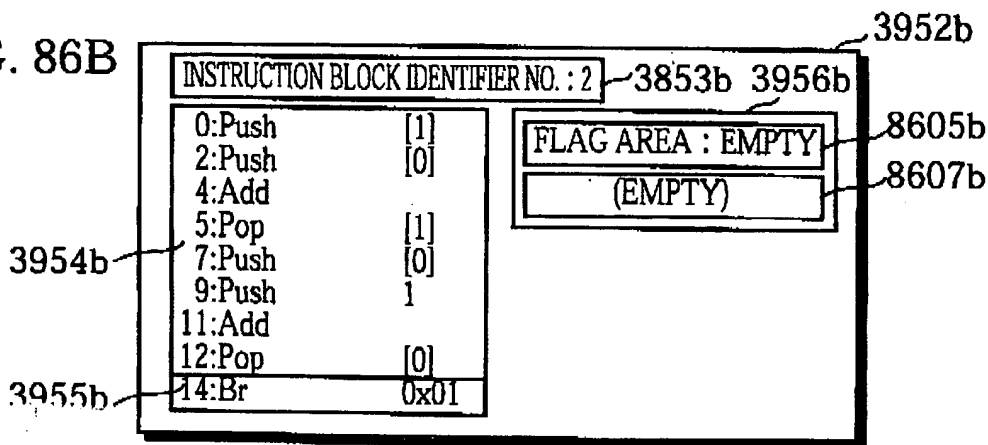


FIG. 86C

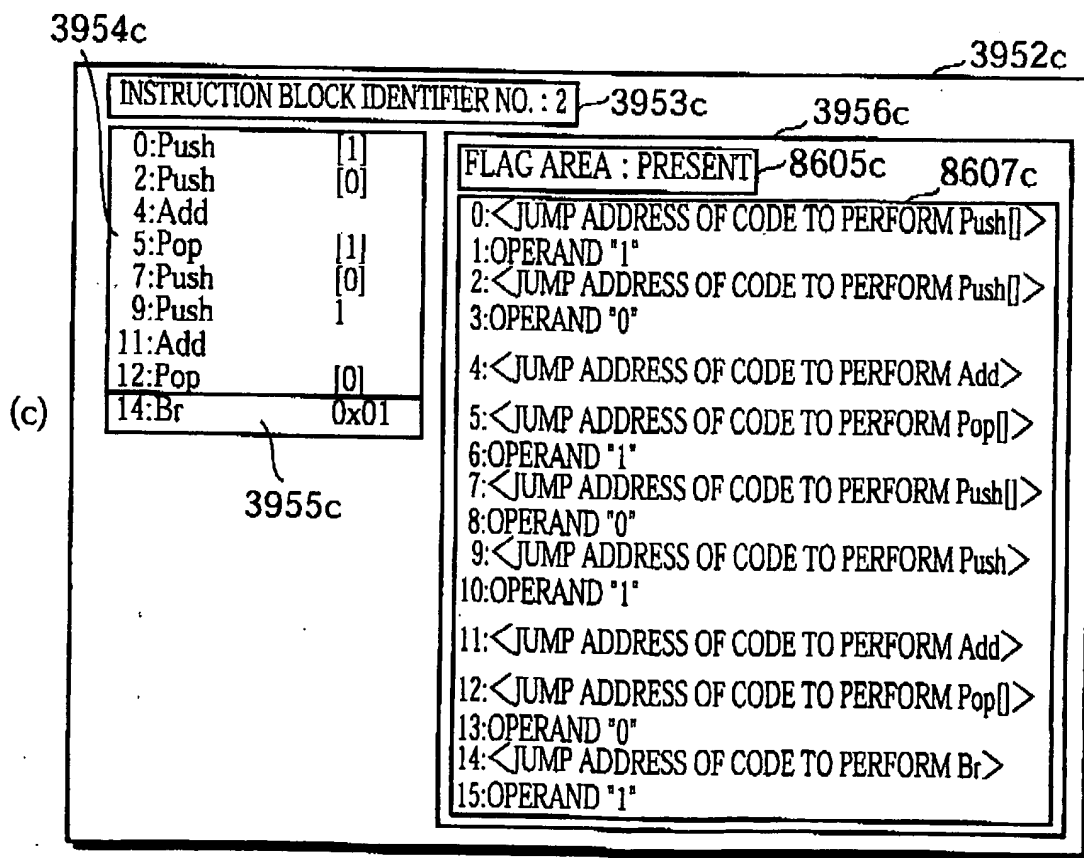


FIG. 87

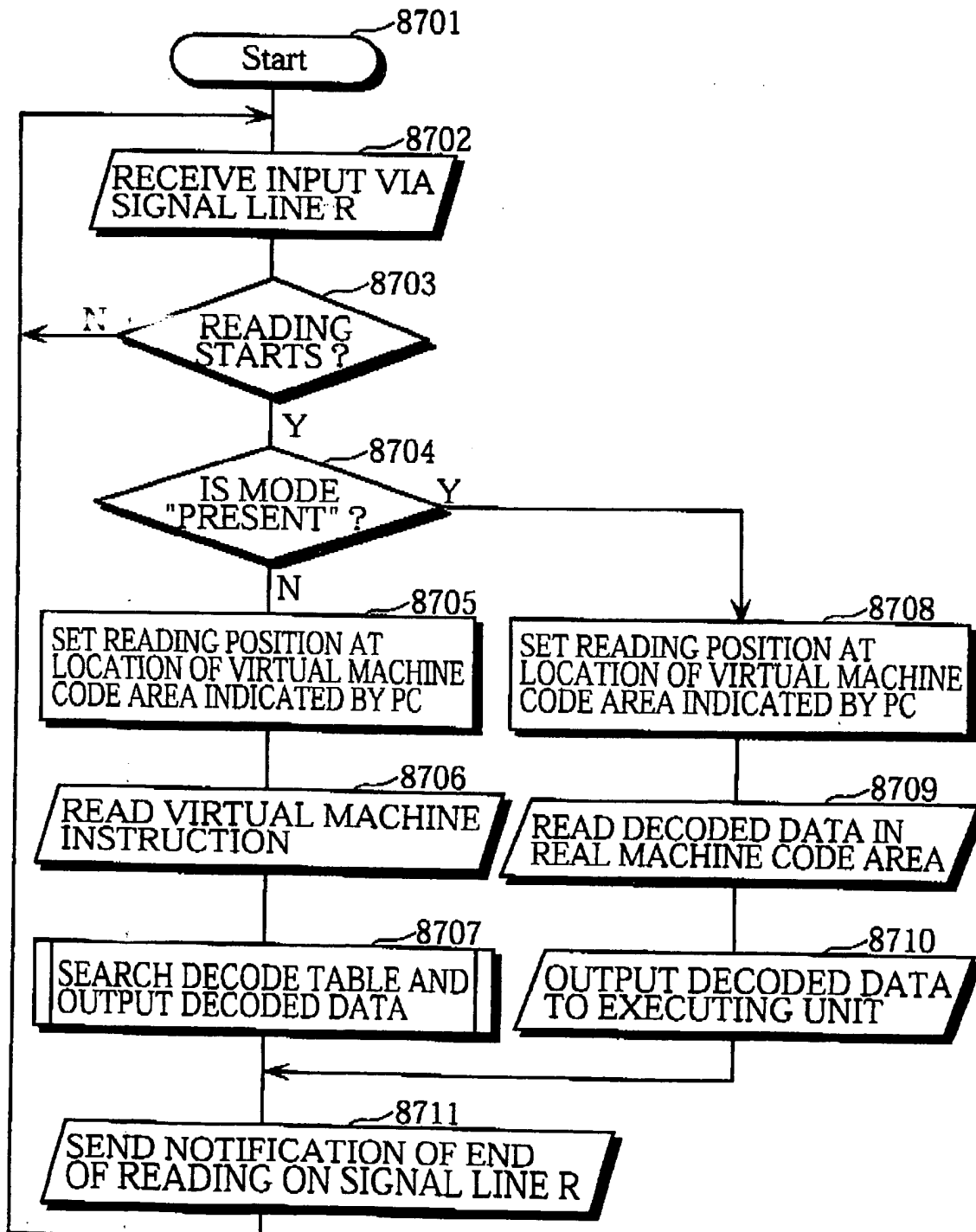


FIG. 88

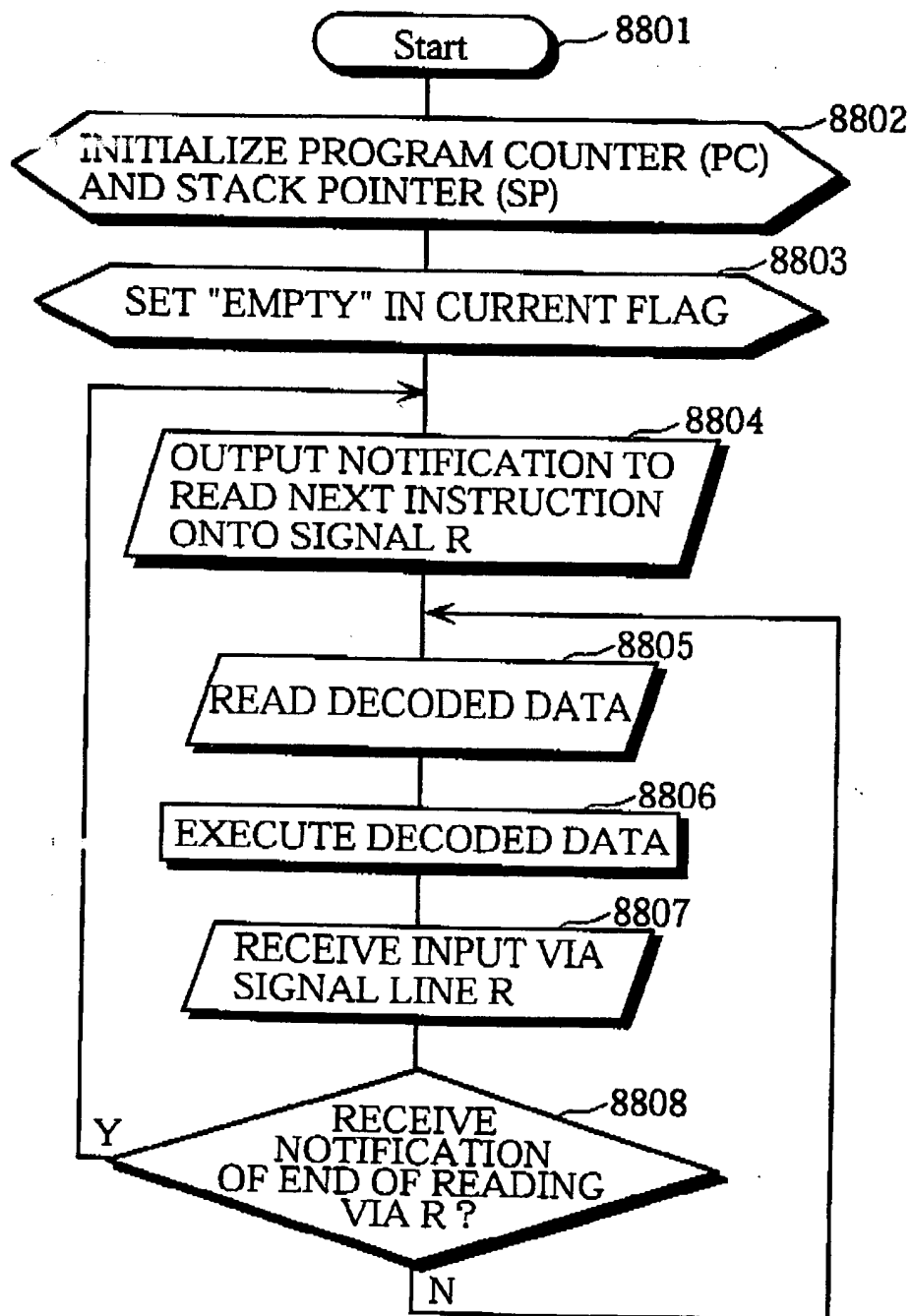


FIG. 89

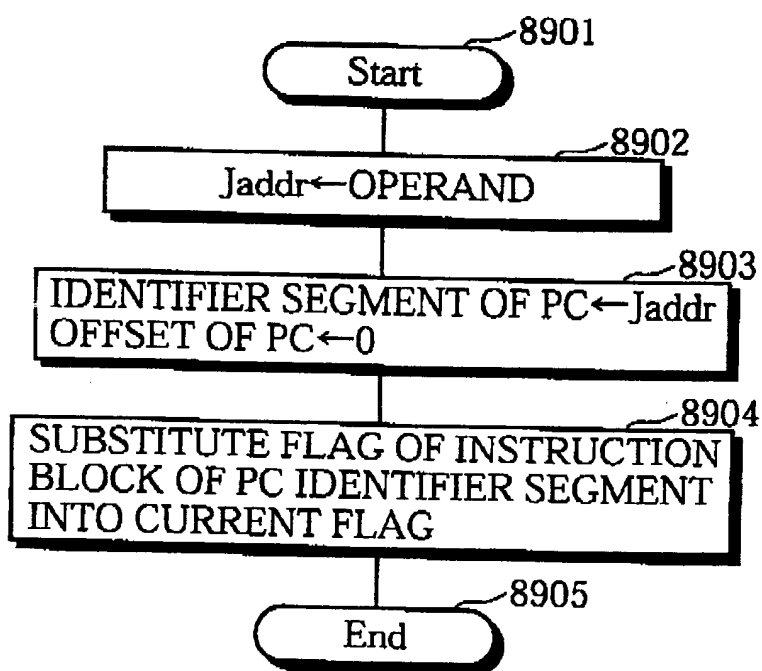


FIG. 90

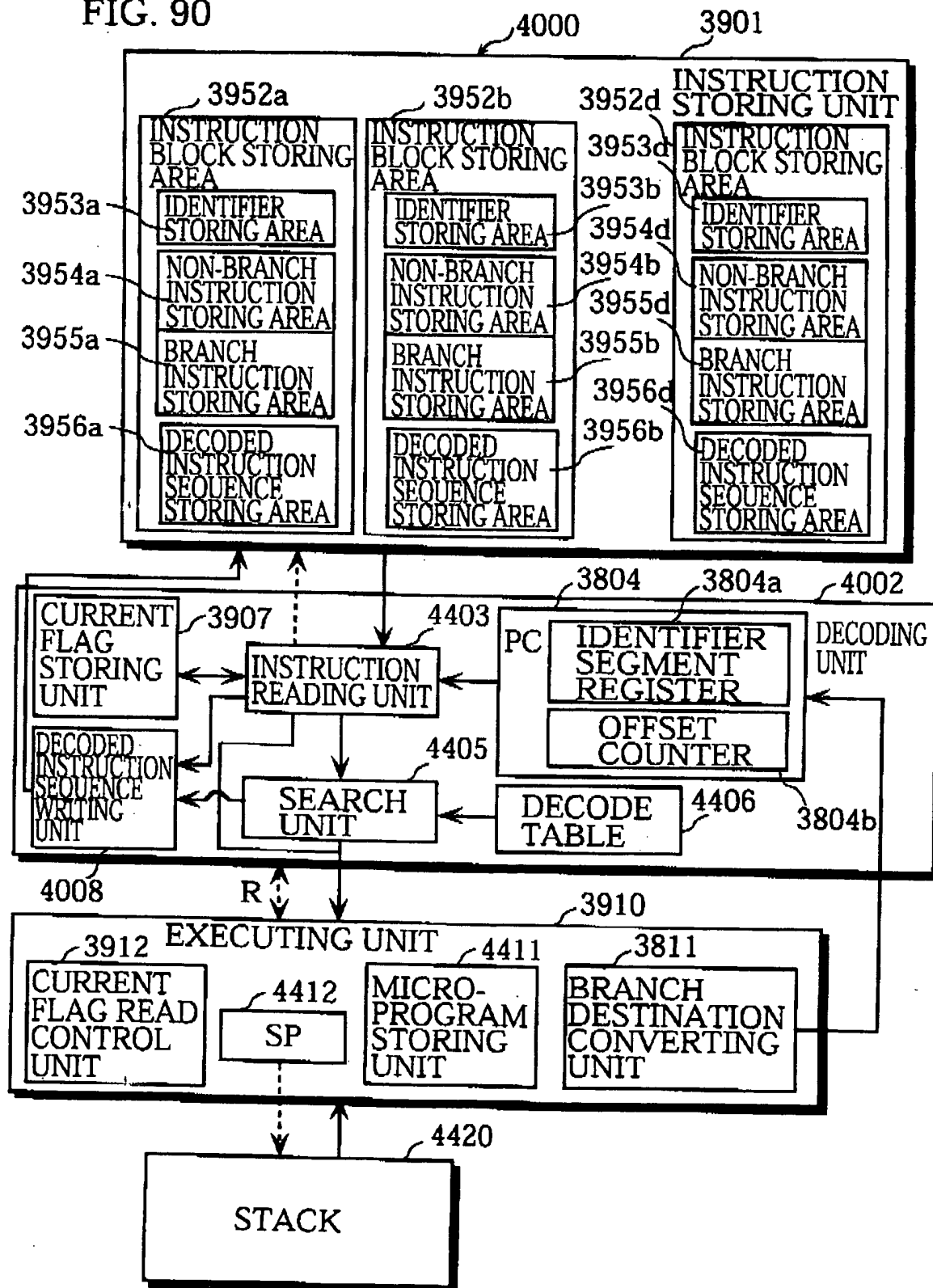


FIG. 91

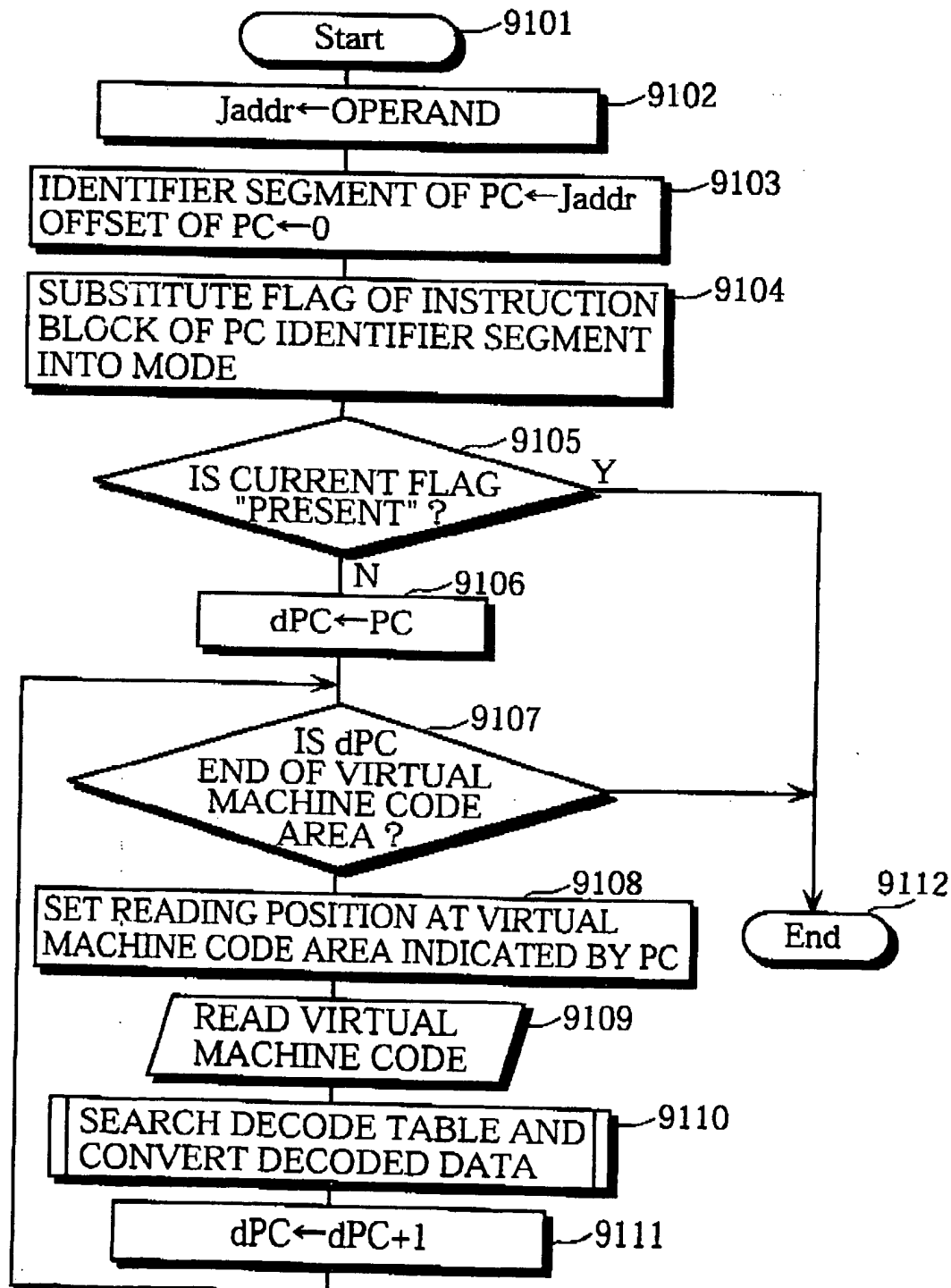


FIG. 92

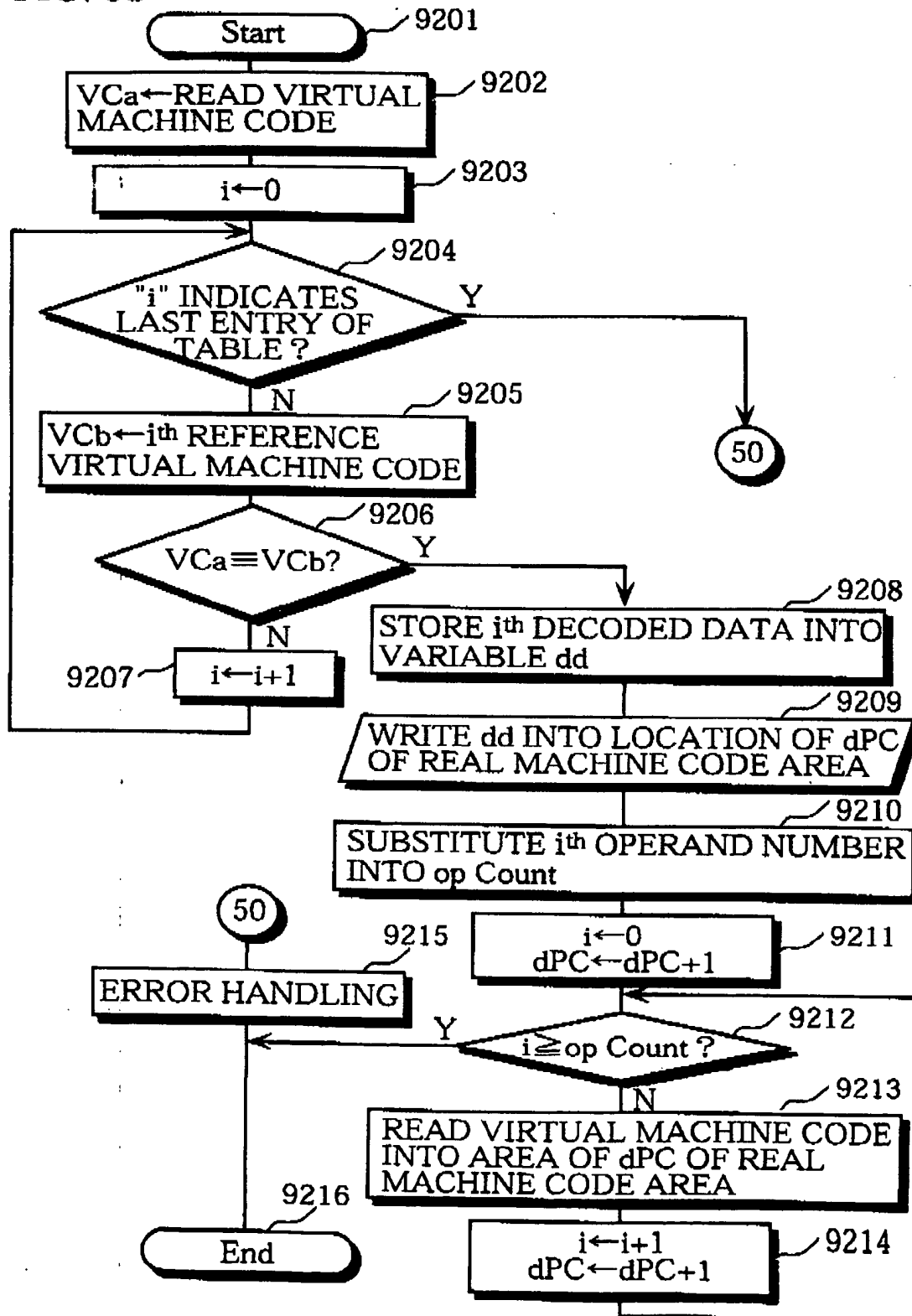




FIG. 93

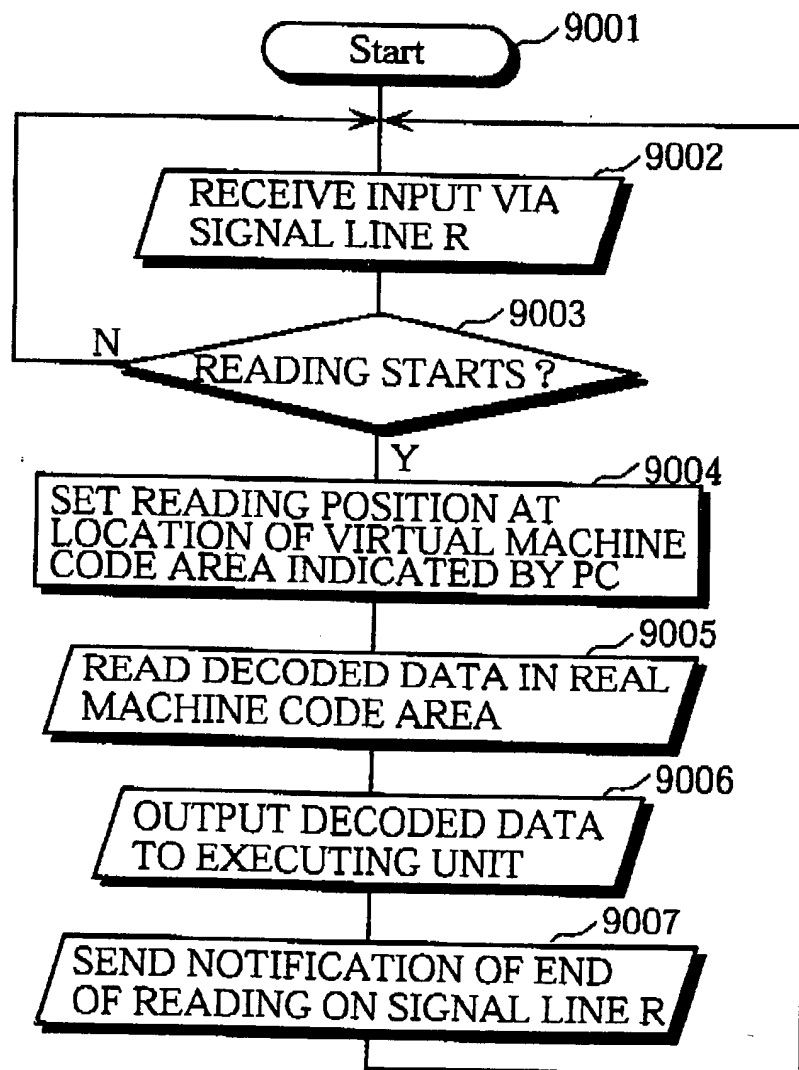


FIG. 94

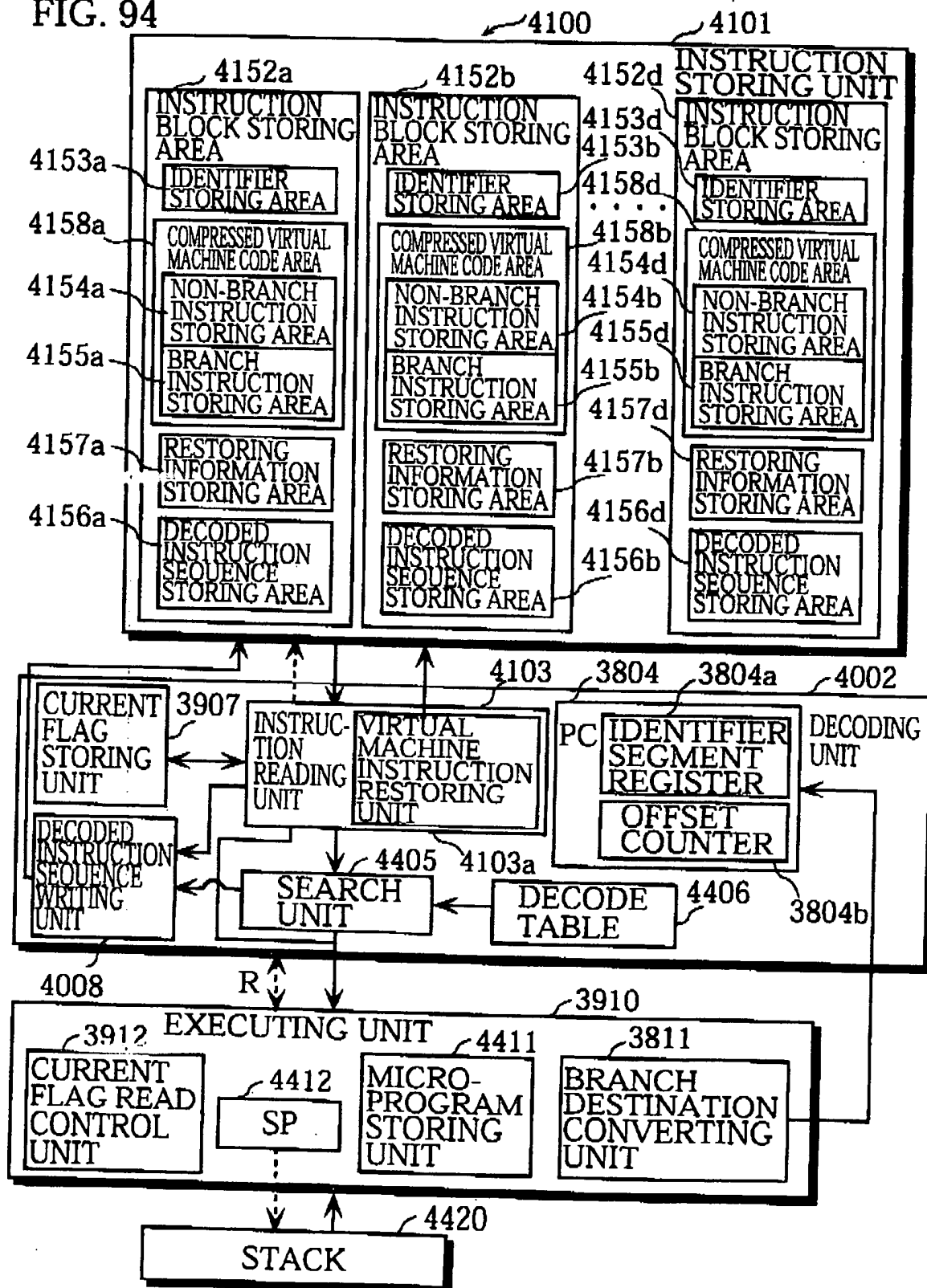


FIG. 95A

BIT SEQUENCE	VIRTUAL MACHINE INSTRUCTION
000	Push [0]
100	Push 0
101	Pop [0]
110	Pop [1]
111	Br 0x01
0011	Stop
0100	Add
0110	Brz 0x03
0111	Br 0x02
00100	Push [1]
00101	Push 1
01010	Push 10
01011	Sub

FIG. 95B

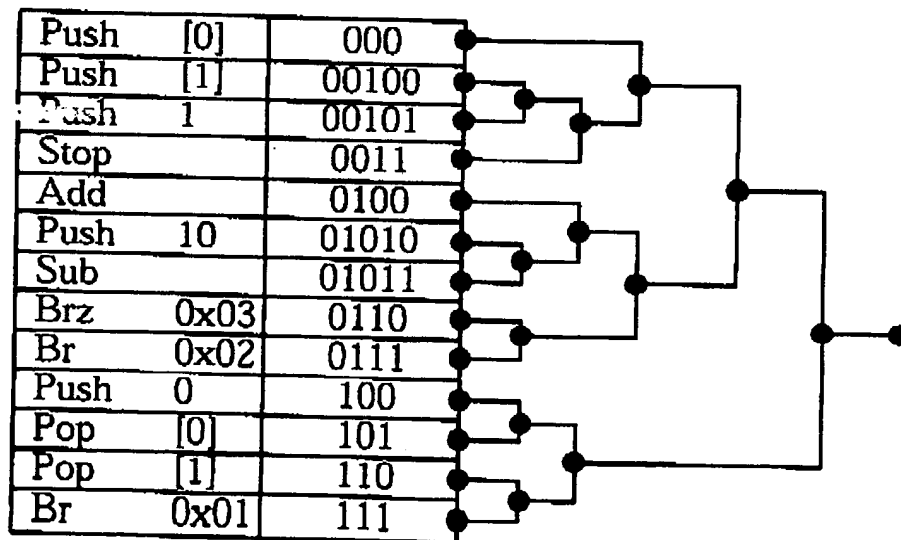


FIG. 96A

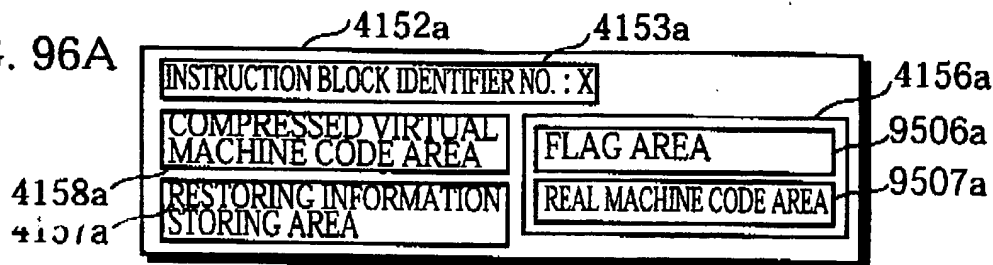


FIG. 96B

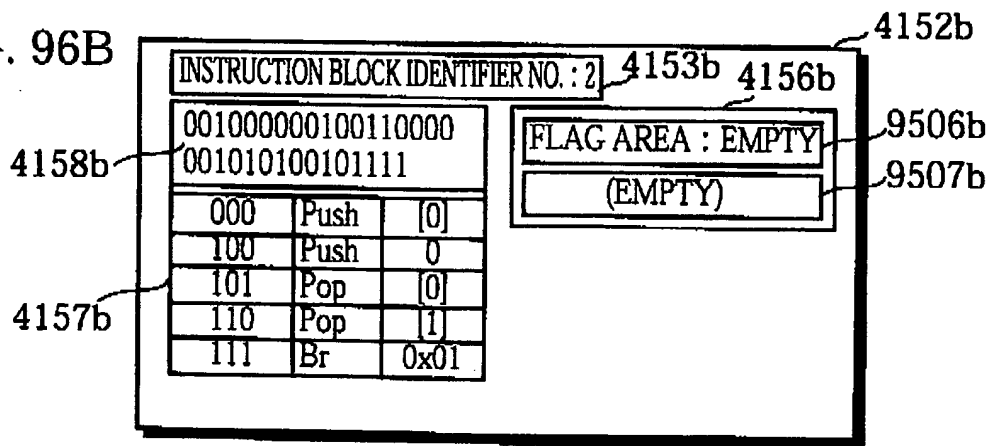


FIG. 96C

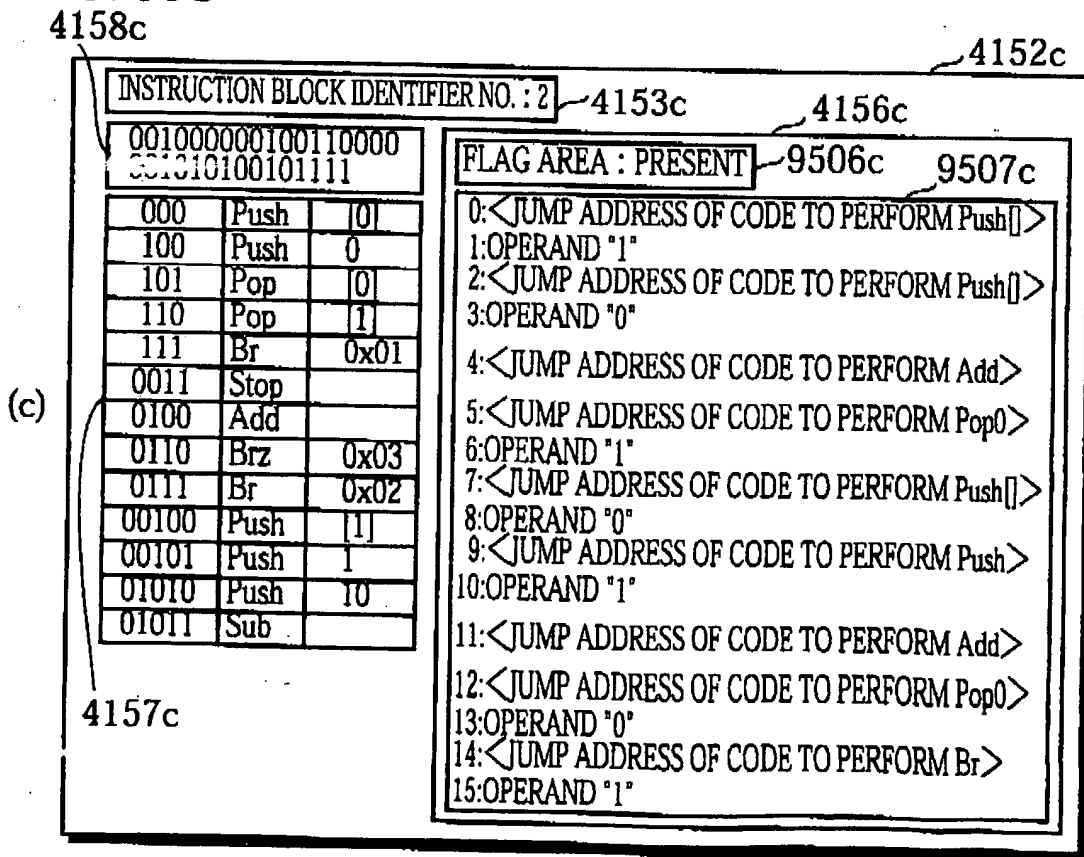


FIG. 97

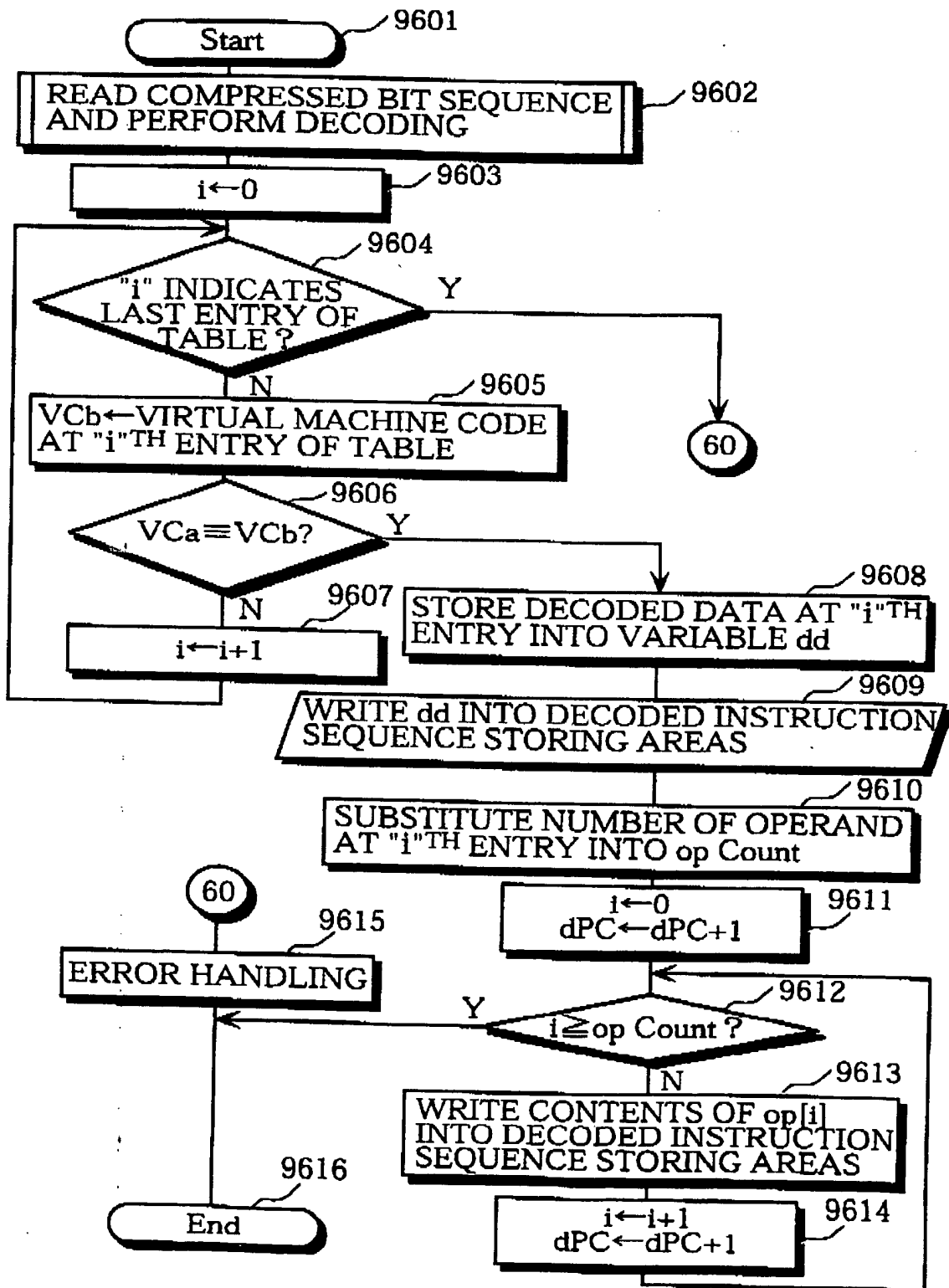


FIG. 98

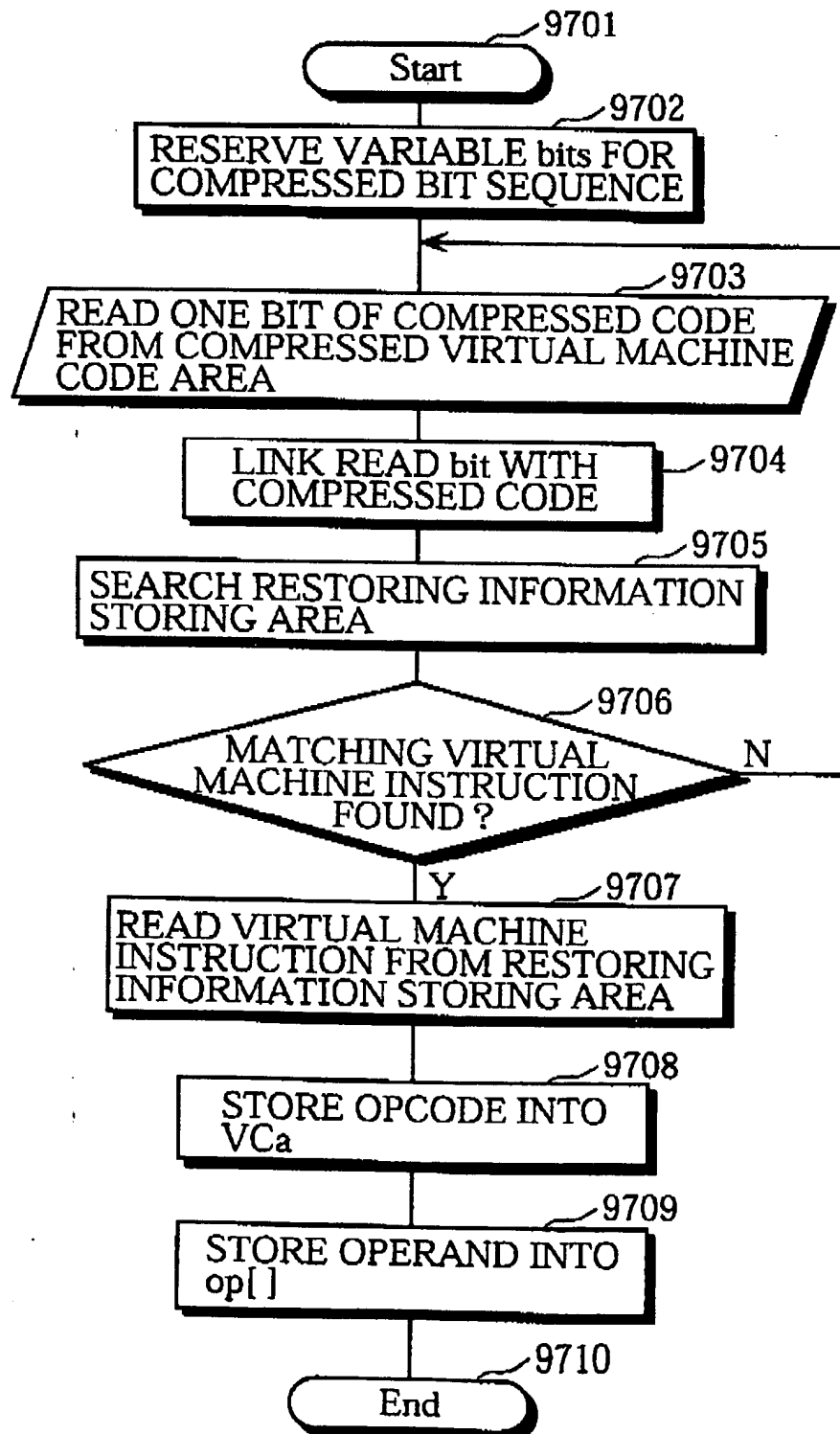


FIG. 99

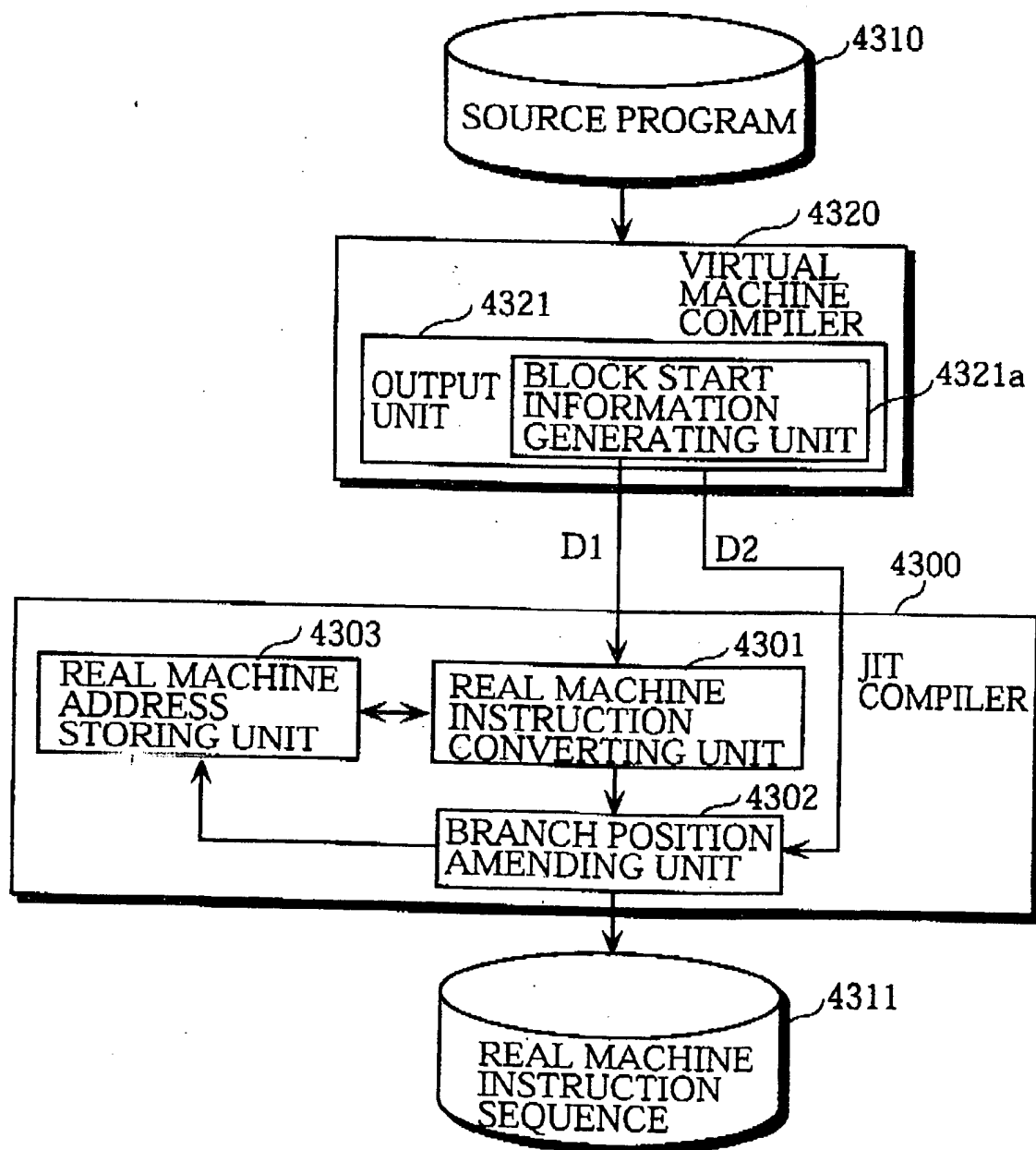


FIG. 100

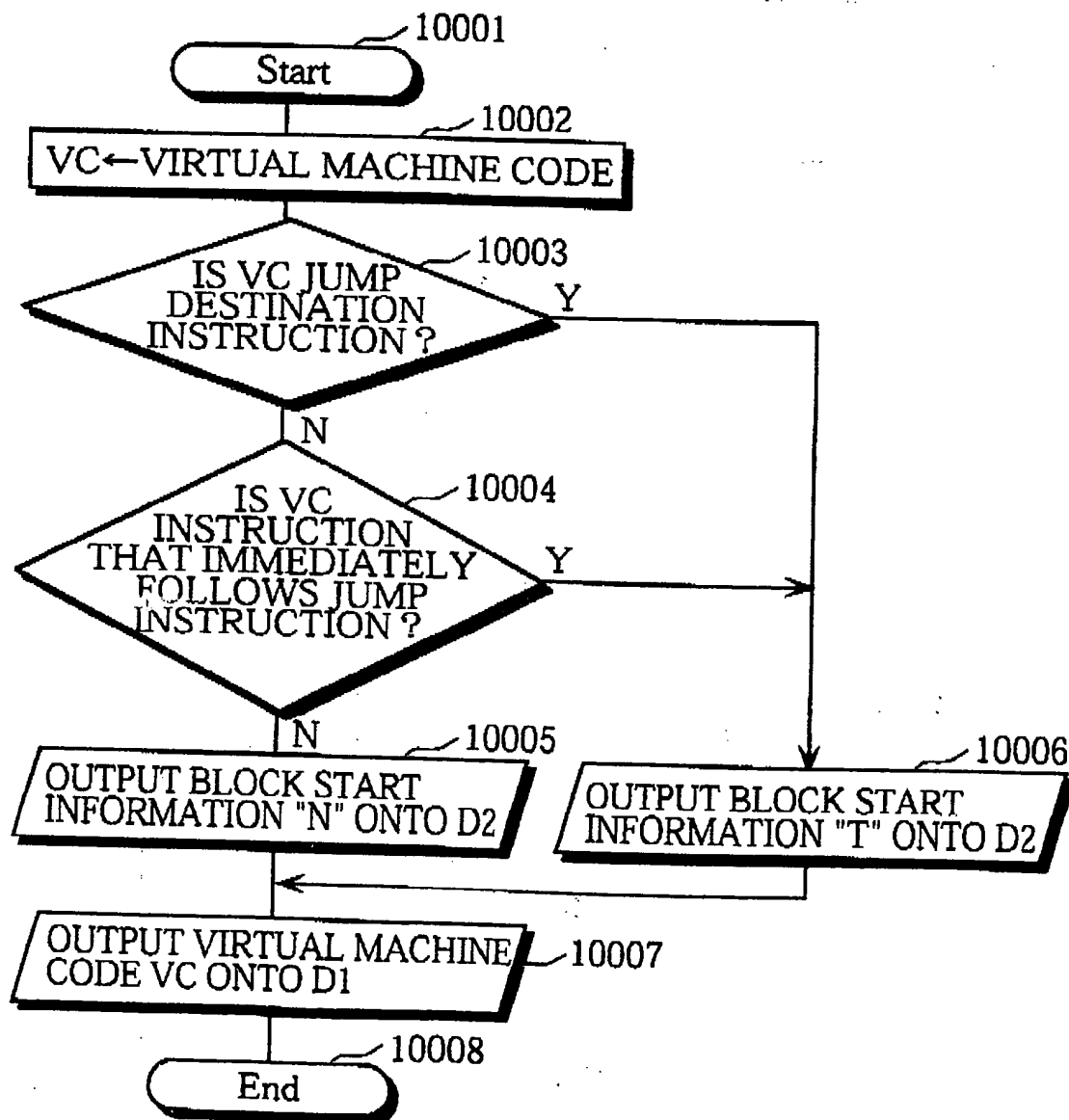




FIG. 101

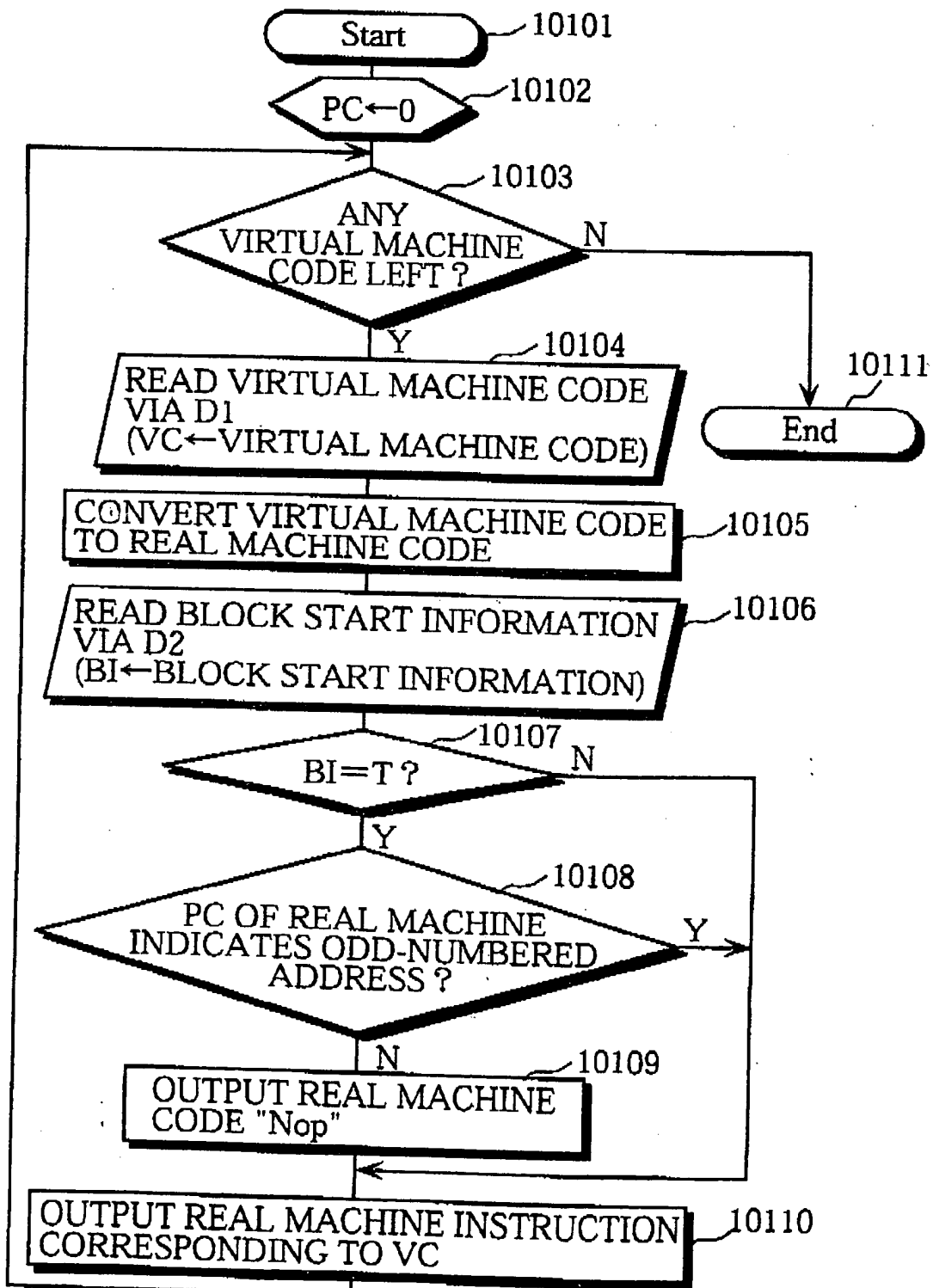


FIG. 102

ADDRESS	VIRTUL MACHINE CODE	REAL MACHINE CODE SIZE	CORRESPONDING REAL MACHINE CODE ADDRESS	BLOCK START INFORMATION	Nop OUTPUT
0	Push 0	4	0-3	T	-
2	Pop [0]	5	4-8	N	-
4	Push 0	4	9-12	N	-
6	Pop [1]	5	13-17	N	-
8	Push [0]	5	18-22	T	-
10	Push 10	4	23-26	N	-
12	Sub	3	27-29	N	-
13	Brz 31	5	30-34	N	Nop
15	Push [1]	5	36-40	T	-
17	Push [0]	5	41-45	N	-
19	Add	3	46-48	N	-
20	Pop [1]	5	49-53	N	-
22	Push [0]	5	54-58	N	-
24	Push 1	4	59-62	N	-
26	Add	3	63-65	N	-
27	Pop [0]	5	66-70	N	-
29	Br 8	3	71-73	N	-
31	Stop	2	74-75	T	-

FIG. 103

U/D	N/T	VIRTUAL MACHINE OPCODE	OPERAND(S)
-----	-----	---------------------------	------------